



**PROGRAMA DE ESTUDIO**  
**PENSAMIENTO COMPUTACIONAL**  
**EDUCACIÓN PARA PERSONAS JÓVENES Y**  
**ADULTAS**  
**FORMACIÓN DIFERENCIADA HUMANISTA-CIENTIFICA**

Nivel 1 o 2 de Educación Media

UCE-MINEDUC  
Marzo 2022

Programa de Estudio Pensamiento Computacional para Educación de Personas Jóvenes y Adultas  
Nivel 1 o 2 de Educación Media.  
Documento aprobado por el Consejo Nacional de Educación mediante el Acuerdo N°019/2022.

Equipo de Desarrollo Curricular  
Unidad de Currículum y Evaluación  
Ministerio de Educación 2022

#### IMPORTANTE

En el presente documento, se utilizan de manera inclusiva términos como “el docente”, “el estudiante”, “el niño”, “el compañero” y sus respectivos plurales (así como otras palabras equivalentes en el contexto educativo) para referirse a hombres y mujeres.

Esta opción obedece a que no existe acuerdo universal respecto de cómo aludir conjuntamente a ambos sexos en el idioma español, salvo usando “o/a”, “los/las” y otras similares, y ese tipo de fórmulas supone una saturación gráfica que puede dificultar la comprensión de la lectura.

## ÍNDICE

Presentación.....	5
Nociones básicas .....	6
Consideraciones generales .....	13
Propósito Formativo .....	29
Enfoque de la asignatura .....	30
Organización curricular.....	31
Objetivos de Aprendizaje.....	31
Orientaciones para el Docente .....	33
Visión panorámica Objetivos de Aprendizaje y conocimientos esenciales Pensamiento Computacional.....	40
Módulo Obligatorio 1 .....	42
Propósito Módulo obligatorio 1.....	43
Ruta de Aprendizaje del Módulo obligatorio 1 .....	44
Actividad de desempeño 1.....	45
Actividad de desempeño 2.....	57
Actividad de desempeño 3.....	70
Actividad de desempeño 4.....	78
Módulo obligatorio 2.....	87
Propósito Módulo obligatorio 2.....	88
Ruta de Aprendizaje del Módulo obligatorio 2 .....	89
Actividad de desempeño 1.....	90
Actividad de desempeño 2.....	101
Actividad de desempeño 3.....	110
Actividad de desempeño 4.....	120
Módulo obligatorio 3.....	127
Propósito Módulo obligatorio 3.....	128
Ruta de Aprendizaje del Módulo 3 .....	129
Actividad de desempeño 1.....	130

Actividad de desempeño 2 .....	140
Actividad de desempeño 3 .....	148
Actividad de desempeño 4 .....	156
Módulo Obligatorio 4 .....	163
Propósito Módulo obligatorio 4.....	164
Ruta de Aprendizaje del Módulo obligatorio 4 .....	165
Actividad de desempeño 1.....	166
Actividad de desempeño 2 .....	172
Actividad de desempeño 3 .....	178
Actividad de desempeño 4.....	183
MÓDULOS ELECTIVOS.....	188
Módulo Aprendizaje basado en Proyectos .....	189
Módulo Aprendizaje basado en Resolución de Problemas.....	197
ANEXO 1 .....	205
ANEXO 2 .....	207

## Presentación

Las Bases Curriculares para EPJA establecen Objetivos de Aprendizaje (OA) de habilidades y actitudes que se integran con conocimientos esenciales para la comprensión de grandes ideas consideradas relevantes en cada asignatura. El presente Programa de estudio es una propuesta de organización curricular que define y desarrolla actividades de desempeño para que los estudiantes construyan los aprendizajes establecidos para cada nivel de enseñanza.

Al Ministerio de Educación le corresponde la tarea de elaborar Programas de estudio que orienten la implementación de las Bases Curriculares para aquellos establecimientos que no han optado por la elaboración de programas propios. Estos programas constituyen un complemento coherente y alineado con las Bases Curriculares y son una herramienta para apoyar a los docentes en el logro de los Objetivos de Aprendizaje y propósitos formativos declarados en cada asignatura y nivel.

Los Programas de estudio constituyen una propuesta que los establecimientos pueden implementar, o ser un referente para aquellos establecimientos que deseen elaborar Programas de estudio propios. En este sentido, responden a las múltiples realidades educativas que se derivan de los distintos contextos en los cuales se imparte la modalidad, y que dan origen a una diversidad de aproximaciones didácticas, metodológicas y organizacionales, que se expresan en el desarrollo de distintos proyectos educativos, todos válidos mientras permitan el logro de los Objetivos de Aprendizaje.

Los Programas de estudio proponen al docente una organización de los Objetivos de Aprendizaje, conocimientos esenciales y grandes ideas de acuerdo con el tiempo disponible dentro del año escolar y constituyen una orientación acerca de cómo desarrollar una comprensión profunda y significativa. Se trata de una estimación temporal aproximada y de carácter propositivo y que, por tanto, puede ser adaptada por los docentes de acuerdo a la realidad de sus estudiantes y de su establecimiento.

Para apoyar la implementación de las bases, los programas proporcionan orientaciones disciplinares, didácticas y criterios de evaluación formativa que pueden utilizarse como apoyo para las actividades de desempeño sugeridas. Las actividades de desempeño son actividades que permiten a los estudiantes poner en “uso” el conocimiento esencial; para lo cual, aplican los procedimientos que definen a las habilidades y actitudes declaradas en los Objetivos de aprendizaje. Las actividades de desempeño, en consecuencia, permiten construir aprendizajes y recoger evidencia de comprensión. Estas actividades se enriquecen con recomendaciones de recursos didácticos complementarios y bibliografía para profesores y estudiantes, y se enmarcan en un modelo pedagógico cuyo enfoque es el de la comprensión, lo que implica establecer conexiones desde la experiencia del estudiante al interior de cada disciplina y también con otras áreas del conocimiento. Las actividades de desempeño de los Programas ilustran un modelo para que cada docente, en su establecimiento, pueda construir nuevas actividades acordes con las diversas realidades.

## Nociones básicas

### OBJETIVOS DE APRENDIZAJE DE HABILIDADES Y ACTITUDES NUCLEARES

Los Objetivos de Aprendizaje definen los aprendizajes terminales esperables para una asignatura determinada en cada nivel escolar y evidencian de forma clara y precisa cuál es el aprendizaje que el estudiante debe lograr. Los Objetivos de Aprendizaje de estas Bases Curriculares refieren a las habilidades y actitudes fundamentales de cada asignatura y se constituyen en el núcleo del aprendizaje.

Las habilidades son definidas como procesos estratégicos centrales para realizar tareas y para solucionar problemas con precisión y adaptabilidad; favorecen la transferencia educativa, es decir, la capacidad para utilizar el conocimiento y aplicarlo en nuevos contextos.

Las actitudes, por su parte, son disposiciones frente a objetos, ideas o personas que incluyen componentes afectivos, cognitivos y valorativos, y que inclinan a las personas a determinados tipos de acciones. Las actitudes que conforman los OA refieren a los cuatro ámbitos del marco de las habilidades para el siglo XXI y su inclusión responde a criterios de pertinencia para ser trabajadas integradamente con las habilidades. En los niveles de Básica, se prioriza el desarrollo de actitudes que fomentan la autonomía y la proactividad, y en los niveles de Media, actitudes que fomentan la responsabilidad personal y social de los estudiantes.

Las actitudes y las habilidades se integran en la construcción de los Objetivos de Aprendizaje nucleares, lo que evidencia su interdependencia y su importancia para una formación integral, que permita a los estudiantes contar con una combinación de valores, disposiciones, habilidades y conocimientos para enfrentar los desafíos del futuro<sup>1</sup>.

### CONOCIMIENTOS ESENCIALES

Los conocimientos esenciales refieren a una red conceptual coherente y rica en conexiones, que permite construir la comprensión sobre los fenómenos y el mundo. El conocimiento entendido como comprensión, permite a los estudiantes refinar, transformar o reemplazar ideas preexistentes que han adquirido en su experiencia vital y cotidiana, y moverse con flexibilidad entre visiones generales y detalles, generalizaciones y ejemplos sobre los fenómenos que estudian.

Los conocimientos esenciales son prioritarios e imprescindibles, pues constituyen una base que permite avanzar de manera progresiva en el aprendizaje de cada asignatura y construir nuevos conocimientos.

---

<sup>1</sup> OECD (2020). Op. Cit., pág. 5.

## PROPÓSITO FORMATIVO

Los propósitos formativos de cada asignatura definen las finalidades educativas que se busca desarrollar a partir de los Objetivos de Aprendizaje y conocimientos esenciales en cada nivel. Entregan el para qué del aprendizaje y buscan evidenciar cómo cada asignatura contribuye al logro de los Objetivos generales de la Educación Media, definidos en la Ley General de Educación.

En estas Bases Curriculares, las grandes ideas operan como propósito formativo de cada nivel, orientando la comprensión y la articulación de los Objetivos de Aprendizaje y los conocimientos esenciales.

## ENFOQUE DE LA ASIGNATURA

Explican los principales principios, teorías y conceptos disciplinares desde los cuales se han construido los aprendizajes de la asignatura. Se presenta una visión actualizada de dichos elementos de acuerdo con el desarrollo de las disciplinas. En el enfoque de la asignatura se explicitan también los énfasis teóricos y perspectivas disciplinares desde las cuales se espera que los docentes y estudiantes aborden los conocimientos, habilidades y actitudes incluidos en los Objetivos de Aprendizaje. Asimismo, en esta sección se explican los enfoques didácticos que permiten orientar la implementación de la asignatura en el aula. Esto último se sustenta en los conceptos, teorías y principios pedagógicos de la enseñanza de cada disciplina.

## HABILIDADES Y ACTITUDES PARA EL SIGLO XXI

La existencia y el uso de la tecnología en el mundo global, multicultural y en constante cambio ha determinado nuevos modos de acceso al conocimiento, de aplicación de los aprendizajes y de participación en la sociedad. Estas necesidades exigen competencias particulares, identificadas internacionalmente como Habilidades del siglo XXI, y responden a los diversos requerimientos del mundo actual, como el aprendizaje de nuevas maneras de pensar, de aprender, de relacionarse con los demás, de comunicarse, de usar la tecnología, de trabajar, de participar en la sociedad, de desarrollarse como persona y de desarrollar la creatividad, entre otros<sup>2</sup>.

Las Habilidades para el siglo XXI corresponden al foco formativo central que propende a la formación integral de los estudiantes. Corresponden a un marco de habilidades y actitudes transversales a todas las asignaturas y a partir de las cuales cada una define sus propios aprendizajes disciplinares. Se presentan organizadas en torno a cuatro ámbitos: Maneras de pensar, Maneras de trabajar, Herramientas para trabajar y Maneras de vivir en el mundo.

---

<sup>2</sup> El conjunto de habilidades seleccionadas para las Bases Curriculares de EPJA corresponden a una adaptación de distintos modelos (Binkley et al., 2012; Fadel et al., 2016) y se han organizado en cuatro categorías: Maneras de pensar, Maneras de trabajar, Herramientas para trabajar y Maneras de vivir en el mundo.

## MANERAS DE PENSAR

### Desarrollo de la creatividad y la innovación

Las personas creativas poseen habilidades de pensamiento divergente, producción de ideas, fluidez, flexibilidad y originalidad. El pensamiento creativo implica abrirse a diferentes ideas, perspectivas y puntos de vista, ya sea en la exploración personal o en el trabajo en equipo. La enseñanza para la creatividad implica asumir que el pensamiento creativo puede desarrollarse en todas las instancias de aprendizaje y en varios niveles: imitación, variación, combinación, transformación y creación original. Por ello, es importante que los docentes consideren que, para lograr la creación original, es necesario haber desarrollado varias habilidades y que la creatividad también puede enseñarse mediante actividades más acotadas según los diferentes niveles.

### Desarrollo del pensamiento crítico

El pensamiento crítico permite discriminar entre informaciones, declaraciones o argumentos, evaluando su contenido y pertinencia. Permite cuestionar la información, tomar decisiones y emitir juicios, como asimismo reflexionar críticamente acerca de diferentes puntos de vista, tanto de los propios como de los demás, ya sea para defenderlos o contradecirlos sobre la base de evidencias. Contribuye así, además, a la autorreflexión y corrección de errores, y favorece la capacidad de estar abierto a los cambios y de tomar decisiones razonadas. El principal desafío en la enseñanza del pensamiento crítico es la aplicación exitosa de estas habilidades en contextos diferentes de aquellos en que fueron aprendidas.

### Desarrollo de la metacognición

Corresponde al concepto de “aprender a aprender”. Se refiere a ser consciente del propio aprendizaje y de los procesos para lograrlo, lo que permite autogestionarlo con autonomía, adaptabilidad y flexibilidad. El proceso de pensar acerca del pensar involucra la reflexión propia sobre la posición actual, fijar los objetivos a futuro, diseñar acciones y estrategias potenciales, monitorear el proceso de aprendizaje y evaluar los resultados. Incluye tanto el conocimiento que se tiene sobre uno mismo como estudiante o pensador, como los factores que influyen en el rendimiento. La reflexión acerca del propio aprendizaje favorece su comunicación, por una parte, y la toma de conciencia de las propias capacidades y debilidades, por otra. Desde esta perspectiva, desarrolla la autoestima, la disciplina, la capacidad de perseverar y la tolerancia a la frustración.

### Desarrollo de Actitudes

- Pensar con perseverancia y proactividad para encontrar soluciones innovadoras a los problemas.
- Pensar con apertura a distintas perspectivas y contextos, asumiendo riesgos y responsabilidades.
- Pensar con consciencia, reconociendo que los errores ofrecen oportunidades para el aprendizaje.
- Pensar con flexibilidad para reelaborar las propias ideas, puntos de vista y creencias.
- Pensar con reflexión propia y autonomía para gestionar el propio aprendizaje, identificando capacidades, fortalezas y aspectos por mejorar.
- Pensar con consciencia de que los aprendizajes se desarrollan a lo largo de la vida y enriquecen la experiencia.
- Pensar con apertura hacia otros para valorar la comunicación como una forma de relacionarse con diversas personas y culturas, compartiendo ideas que favorezcan el desarrollo de la vida en sociedad.



## MANERAS DE TRABAJAR

### Desarrollo de la comunicación

La comunicación, ya sea escrita, oral o multimodal, requiere generar estrategias y herramientas que se adecuen a diversas situaciones, propósitos y contextos socioculturales, con el fin de transmitir lo que se desea de manera efectiva. La comunicación permite desarrollar la empatía, la autoconfianza, la valoración de la interculturalidad, así como la adaptabilidad, la creatividad y el rechazo a la discriminación.

### Desarrollo de la colaboración

La colaboración entre personas con diferentes habilidades y perspectivas faculta al grupo para tomar mejores decisiones que las que se tomarían individualmente. Además, el trabajo colaborativo entre pares determina nuevas formas de aprender y de evaluarse a sí mismo y a los demás, lo que permite visibilizar los modos en que se aprende; esto conlleva nuevas maneras de relacionarse en torno al aprendizaje.

La colaboración implica, a su vez, actitudes clave para el aprendizaje en el siglo XXI, como la responsabilidad, la perseverancia, la apertura de mente hacia lo distinto, la aceptación y valoración de las diferencias, la autoestima, la tolerancia a la frustración, el liderazgo y la empatía.

### Desarrollo de Actitudes

- Trabajar colaborativamente en la generación, desarrollo y gestión de proyectos y la resolución de problemas, integrando las diferentes ideas y puntos de vista.
- Trabajar con responsabilidad y liderazgo en la realización de las tareas colaborativas y en función del logro de metas comunes.
- Trabajar con empatía y respeto en el contexto de la diversidad, eliminando toda expresión de prejuicio y discriminación.
- Trabajar con autonomía y proactividad en trabajos colaborativos e individuales para llevar a cabo eficazmente proyectos de diversa índole.

## HERRAMIENTAS PARA TRABAJAR

### Desarrollo de la alfabetización digital

Promueve el desarrollo del pensamiento computacional, la autonomía y el trabajo en equipo, la creatividad, la participación en redes de diversa índole, y el interés por ampliar los propios intereses y horizontes culturales, por medio del uso responsable de la tecnología para hacer frente a nuevos desafíos, como la ciberseguridad y el autocuidado. La utilización de la tecnología como herramienta de trabajo implica dominar las posibilidades que ofrece, como asimismo darle un uso creativo e innovador que, a la vez, promueva el pensamiento crítico. A partir de esto, la alfabetización digital apunta también a la resolución de problemas en el marco de la cultura digital que caracteriza al siglo XXI, aprovechando las herramientas que nos da la programación, el pensamiento computacional, la robótica e internet, entre otros, para desarrollar habilidades que permitan crear contenidos digitales, informarnos a partir de la tecnología y vincularnos con los demás utilizando la tecnología.

### Desarrollo del uso de la información

Dice relación con la eficacia y eficiencia en la búsqueda, el acceso, el procesamiento, la clasificación, la integración, la gestión, la evaluación crítica, el uso creativo y ético, y la comunicación, de la información. Implica formular preguntas, indagar y generar estrategias para seleccionar, organizar y comunicar la información. Tiene además siempre en cuenta tanto los aspectos éticos y legales que la regulan, como el respeto a los demás y a su privacidad. Promueve también el acceso, uso responsable, aplicación eficaz y evaluación crítica de Tecnologías de Información y Comunicación (TIC), y su uso creativo de acuerdo con distintos propósitos, atendiendo a las características y convenciones de diversos contextos multiculturales.

### Desarrollo de Actitudes

- Aprovechar las herramientas disponibles para aprender y resolver problemas.
- Interesarse por las posibilidades que ofrece la tecnología para el desarrollo intelectual, personal y social del individuo.
- Valorar las TIC como una oportunidad para informarse, investigar, socializar, comunicarse y participar como ciudadano.
- Actuar responsablemente al gestionar el tiempo para llevar a cabo eficazmente los proyectos personales, académicos y laborales.
- Actuar de acuerdo con los principios de la ética en el uso de la información y de la tecnología, respetando la propiedad intelectual y la privacidad de las personas.

## MANERAS DE VIVIR EN EL MUNDO

### **Desarrollo de la ciudadanía local y global**

La ciudadanía se refiere a la participación del individuo en su contexto desde una perspectiva política, social, territorial, cultural, económica, medioambiental, entre otras dimensiones. Por ello, es necesaria la interacción eficaz con las instituciones públicas y la participación en iniciativas que apoyen la cohesión social. La participación también implica reflexionar y tener un juicio crítico acerca de los mensajes de los medios de comunicación masiva, de modo de adoptar una postura razonada ante ellos. La conciencia de ser ciudadano promueve el sentido de pertenencia y la valoración y ejercicio de los principios democráticos, como los derechos humanos y la igualdad, así como asumir sus responsabilidades como tal. En este sentido, el respeto a los demás, a su privacidad, y a las diferencias valóricas, religiosas y étnicas cobra gran relevancia; se relaciona directamente con una actitud empática, de mentalidad abierta y de adaptabilidad.

### **Desarrollo del plan de vida y carrera**

La construcción y consolidación de un proyecto de vida y de una carrera, oficio u ocupación, requiere la capacidad de adaptarse a los cambios para poder desenvolverse en distintos roles y contextos. Para el logro de objetivos personales, es necesario establecer metas, crear estrategias para conseguirlas, desarrollar la autogestión, actuar con iniciativa y compromiso, ser autónomo para ampliar los aprendizajes, ser autocrítico, reflexionar críticamente y estar dispuesto a integrar las retroalimentaciones recibidas. Por otra parte, para lograr estas metas se requiere interactuar con los demás de manera flexible, con la capacidad de trabajar en equipo para la búsqueda de soluciones. Esto permite el desarrollo de liderazgo, responsabilidad, ejercicio ético del poder y el respeto por las diferencias en ideas y valores.

### **Desarrollo de responsabilidad personal y social**

La responsabilidad personal y social se interrelacionan constantemente. En lo personal, el respeto por los demás y el rechazo a la discriminación, la conciencia acerca de la propia cultura y las relaciones de esta con las del mundo, el compromiso con la propia vida y el contexto inmediato, y el control de la agresión, la violencia y la autodestrucción permiten que las personas se desarrollen de una manera integral. Por otra parte, el compromiso con la propia persona se traduce, a su vez, en una manera sana y activa de relacionarse con los demás, generando confianza en los otros y comunicándose de una manera asertiva, empática, libre de prejuicios, que acepte los distintos puntos de vista y contribuyendo a mejorar la sociedad en la que vive. Estas habilidades apuntan a ser consciente de sí mismo y de los otros, y a realizar acciones concretas que den cuenta de la responsabilidad que tiene el individuo con su vida y con su entorno.

### Desarrollo de Actitudes

- Perseverar en torno a metas con miras a la construcción de proyectos de vida y al aporte a la sociedad y al país con autodeterminación, autoconfianza y respeto por uno mismo y por los demás.
- Participar asumiendo posturas razonadas en distintos ámbitos: cultural, social, político, medioambiental, entre otros.
- Tomar decisiones razonadas y que contribuyan al bien común, respetando los derechos humanos, la diversidad y la multiculturalidad.
- Actuar con honestidad, responsabilizándose de las propias acciones y decisiones con consciencia, y de las implicancias que estas tienen sobre uno mismo y los otros.

## Consideraciones generales

Las consideraciones que se presentan a continuación son relevantes para una óptima implementación de los Programas de Estudio, se vinculan estrechamente con los enfoques curriculares, y permiten abordar de mejor manera los Objetivos de Aprendizaje de las Bases Curriculares.

### El estudiante de Educación para Jóvenes y Adultos

#### PERFIL DE EGRESO

La formación habilita al estudiante para conducir su propia vida en forma autónoma, plena y responsable, de modo que pueda desarrollar planes de vida y proyectos personales, continuar su proceso educativo formal mediante la educación superior, o incorporarse a la vida laboral.

Los estudiantes que egresan de la modalidad de Jóvenes y Adultos han desarrollado los conocimientos, habilidades y actitudes definidas en el currículum nacional y transfieren sus aprendizajes a distintos ámbitos: social, cultural, cívico, laboral, intelectual y personal. A partir de dichos aprendizajes, son capaces de alcanzar sus metas académicas y laborales, y de construir un proyecto de vida de acuerdo con sus necesidades e intereses, actuando con autonomía, responsabilidad.

Considerando el marco de Habilidades del siglo XXI y los Objetivos generales de la Ley General de Educación, las Bases Curriculares para la EPJA definen un conjunto de diez competencias que reúnen habilidades, actitudes y conocimientos que los estudiantes han adquirido al finalizar el segundo nivel de Educación Media de la modalidad. Estas competencias se organizan según los ámbitos de las Habilidades del siglo XXI, y su relación de tributación con las habilidades y actitudes nucleares de los Objetivos de Aprendizaje. La competencia 1 se refiere al dominio disciplinar de las asignaturas que los estudiantes deberán dominar al finalizar la Educación Media.

#### **Dominio disciplinar**

1. Aplica conocimientos y habilidades disciplinares de las áreas del lenguaje, las matemáticas, las ciencias, la historia y la geografía y el idioma extranjero inglés en contextos que impliquen aprendizaje y desarrollo personal.

#### **Maneras de pensar**

2. Gestiona el proceso de aprendizaje personal por medio de habilidades de metacognición, reflexión y comunicación, demostrando autonomía, motivación y una sólida autoestima y confianza en las propias capacidades para mejorar y enriquecer su desarrollo personal y cognitivo.
3. Identifica problemas, elabora argumentos, considera nuevas ideas, y propone soluciones creativas e innovadoras ante los desafíos que enfrenta.

4. Piensa de manera crítica y elabora puntos de vista y opiniones propias, utilizando evidencia y con una actitud abierta, dispuesta a cuestionar los supuestos y a reconsiderar las propias visiones.

### **Maneras de trabajar**

5. Trabaja de manera colaborativa con otros en la resolución de problemas y en el desarrollo de proyectos, demostrando habilidades interpersonales de comunicación, gestión y monitoreo del trabajo, y capacidad para asumir roles, reconocer fortalezas y aceptar debilidades, y una actitud perseverante para alcanzar los objetivos propuestos.
6. Se comunica efectivamente con otros en lengua materna y en una lengua extranjera, con diferentes propósitos y en diversos contextos, por medio de habilidades de comunicación oral, escrita y no verbal, demostrando capacidad de escuchar y comprender distintos mensajes, y una valoración positiva del lenguaje como fuente de enriquecimiento cultural y personal.

### **Herramientas para trabajar**

7. Utiliza internet y las herramientas digitales de manera efectiva y eficiente, demostrando habilidades de búsqueda, selección, manejo y producción de información, y capacidad para resolver tareas, reconociendo los aspectos éticos y legales involucrados en el acceso y uso de la información en ambientes digitales.
8. Demuestra compromiso y capacidad de autogestionar el aprendizaje en las diversas instancias de formación que enfrenta, por medio de habilidades que le permitan desenvolverse en distintos roles y contextos y planificar un proyecto de vida personal y laboral en el tiempo, desarrollando una disposición favorable al aprendizaje a lo largo de la vida.

### **Maneras de vivir en el mundo**

9. Se relaciona de manera respetuosa, empática y constructiva con otros en las diversas instancias de intercambio y colaboración que enfrenta, demostrando conciencia y reconocimiento de la propia cultura y la de los demás, y una actitud de rechazo a la violencia, a la agresión y a la discriminación.
10. Demuestra conciencia de los derechos y responsabilidades ciudadanas al relacionarse con sus pares, con la comunidad y con las instituciones públicas, practicando habilidades de interacción eficaz, de participación y toma de decisiones, mostrando un compromiso con el bien común, la cohesión social, los Derechos Humanos y los principios de la democracia, a nivel local y global.

## CONTEXTUALIZACIÓN CURRICULAR

La contextualización curricular es el proceso de apropiación y desarrollo del currículum en una realidad educativa concreta. Este se lleva a cabo considerando las características particulares del contexto escolar (por ejemplo, el medio en que se sitúa el establecimiento educativo, la cultura, el proyecto educativo institucional de la escuela y la comunidad escolar, el tipo de formación diferenciada que se imparte - humanístico-científica o técnico profesional), lo que posibilita que el proceso educativo adquiera significatividad para los estudiantes desde sus propias realidades y facilita, así, el logro de los Objetivos de Aprendizaje.

El marco de habilidades y actitudes que define esta propuesta permite desarrollar actitudes y habilidades que facilitan formas de pensar, de vivir en el mundo, formas de trabajar y herramientas para trabajar que definen el perfil del estudiante EPJA y que pueden ser utilizados como estrategias para atender a las necesidades de contextualización las diferencias que se presenten en las aulas. Los Programas de estudio son una propuesta de diseño de clases, de actividades y de evaluaciones flexible, que pueden modificarse, ajustarse y transferirse a diferentes realidades y contextos, considerando, entre otros:

**Diversidad etaria;** debido a que la edad de los estudiantes de Educación para Jóvenes y Adulto puede variar de los 15 a más de 50 años de edad, las actividades propuestas se han diseñado desde un principio de flexibilidad que permita en las aulas ajustarse a las distintas necesidades y posibilidades de estudiantes que no han iniciado o interrumpido su trayectoria formativa por un corto o un largo período de tiempo.

**Tipos de establecimientos:** considerando las distintas posibilidades originadas por el tipo de establecimiento en las que se implementa la modalidad; tercera jornada, centros de educación integrada de adultos, establecimientos educacionales al interior de recintos penitenciarios y al interior de unidades militares, la ilustración didáctica de las actividades propuestas en el programa sugiere el uso de recursos y procedimientos tanto análogos como virtuales.

**Trayectorias formativas:** considerando que por razones diversas las trayectorias formativas de los estudiantes EPJA se interrumpen, y en consecuencia, la progresión de aprendizajes de las asignaturas que forman parte del Plan de Formación General en sus distintos Niveles de Educación Básica y Educación Media: Lenguaje y Comunicación/Lengua y Literatura, Matemática, Ciencias e Historia, Geografía, Ciencias Sociales y Educación Ciudadana, pueden estar afectadas, de modo que para la implementación de los programas de estudio se necesite realizar procesos previos de nivelación que permitan a los estudiantes avanzar en su trayectoria formativa. La implementación del programa se ha diseñado en un tiempo estimativo que, de acuerdo al plan de estudio, puede ajustarse a las necesidades formativas de los estudiantes.

## INCLUSIÓN Y DIVERSIDAD

En el trabajo pedagógico, es importante comprender que la diversidad se entiende en términos culturales, sociales, étnicos, religiosos, de género, de estilos de aprendizaje y de niveles de conocimiento y/o de trayectorias escolares. Esta diversidad enriquece los escenarios de aprendizaje y está asociada a los siguientes desafíos:

- Desarrollar aprendizajes significativos que se relacionen con el contexto y la realidad de los estudiantes.
- Generar oportunidades inclusivas para desarrollar el aprendizaje en todos los estudiantes.
- Favorecer y potenciar metodologías integradoras y colaborativas tales como Aprendizaje Basado en Proyectos (ABP) y Aprendizaje Basado en la Resolución de Problemas (ABRP).

Atender a la diversidad de estudiantes, en sus contextos, implica reconocer las necesidades educativas de los estudiantes para diseñar experiencias de aprendizaje considerando tiempos, recursos y estrategias para que cada estudiante logre un aprendizaje de calidad. La experiencia y conocimiento que tengan los docentes sobre su asignatura y las estrategias que promuevan un aprendizaje profundo, son herramientas para tomar decisiones pertinentes y oportunas respecto de las necesidades de sus alumnos.

Para los estudiantes con necesidades educativas especiales, el conocimiento de los profesores, el apoyo y las recomendaciones de los especialistas contribuyen a que todos desarrollen al máximo sus capacidades. Algunas orientaciones para considerar:

- Generar ambientes de aprendizaje inclusivos, lo que implica que cada estudiante debe sentir seguridad para participar, experimentar y contribuir de forma significativa a la clase. Se recomienda destacar positivamente las características particulares y rechazar toda forma de discriminación, agresividad o violencia.
- Proveer igualdad de oportunidades, asegurando que los estudiantes puedan participar por igual en todas las actividades, evitando asociar el trabajo de aula con estereotipos asociados a género, características físicas o cualquier otro tipo de sesgo que provoque discriminación.
- Utilizar diversos materiales, estrategias didácticas y actividades que se adecuen a las singularidades de los estudiantes y sus intereses.
- Promover un trabajo sistemático, con actividades variadas para diferentes estilos de aprendizaje y con ejercitación abundante, procurando que todos tengan acceso a oportunidades de aprendizaje enriquecidas.



## Orientaciones pedagógicas Programas de estudio EPJA

Todas las actividades siguen los pasos que caracterizan el proceso de aprendizaje en los jóvenes y adultos: identificar la necesidad del aprendizaje; crear una estrategia y recursos para alcanzarlos; desarrollar la estrategia y evaluarla. Para aprender, necesitan saber cuál es el propósito de su aprendizaje, aplicar lo aprendido en la vida profesional; y ser agentes de su propio aprendizaje, utilizando su experiencia.

La etapa inicial del aprendizaje es de gran importancia, ya que, si bien el estudiante puede no estar siempre consciente de lo que necesita aprender, la motivación y el compromiso por el aprendizaje como un medio para adquirir autonomía y aprender a aprender, pueden operar como incentivos poderosos para encontrar un sentido al aprendizaje escolar. Asimismo, es relevante que los estudiantes participen en el proceso de diseño del aprendizaje. La literatura señala que, en los estudiantes adultos, compartir el control de las estrategias de aprendizaje lo hace más eficaz.<sup>3</sup> Hacer participar a los estudiantes adultos como agentes de su aprendizaje, satisface su necesidad de conocer y estimula su autoconcepto como alumnos independientes<sup>4</sup>.

### Organización modular del Programa de estudio

Los Programas de estudio para las Bases Curriculares de la Educación de Jóvenes y Adultos, proponen una estructura modular que organiza los Objetivos de Aprendizaje de habilidades y actitudes, los conocimientos esenciales y las grandes ideas de cada asignatura de acuerdo con las Bases Curriculares aprobadas para la modalidad.

Los módulos se definen como bloques unitarios de aprendizaje que integran habilidades, actitudes y conocimientos requeridos para adquirir desempeños flexibles en una determinada área o asignatura.

Todas las asignaturas, tanto del plan de Formación General como de Formación Instrumental cuentan con Programas de estudio modulares para su implementación. En cuanto a la estructura, cada asignatura se organiza por nivel en cuatro módulos obligatorios y cuatro módulos electivos. Los módulos obligatorios organizan los Objetivos de Aprendizaje, conocimientos esenciales y grandes ideas de cada nivel, y los módulos electivos ofrecen oportunidades de profundizar en el desarrollo del OA y en la comprensión de las grandes ideas del nivel, por medio del desarrollo de proyectos o la resolución de problemas.

---

<sup>3</sup> Knowles, M. S., Holton III, E. F., & Swanson, R. A. (2014). The adult learner: The definitive classic in adult education and human resource development. Routledge, pág. 148.

<sup>4</sup> Ibídem.

## MÓDULOS OBLIGATORIOS:

En coherencia con las Bases Curriculares, los módulos obligatorios organizan los Objetivos de Aprendizaje, los conocimientos esenciales y las grandes ideas del nivel. Cada módulo presenta cuatro actividades de aprendizaje y evaluación que desarrollan, como foco principal, las habilidades y actitudes de los Objetivos de Aprendizaje del nivel. En las Bases Curriculares para EPJA, las habilidades son entendidas como conocimientos procedimentales que desarrollan destrezas de pensamiento y hábitos de mente que permiten pensar en los contenidos en profundidad. Desarrollar habilidades permite a los estudiantes aprender a pensar sobre el conocimiento, ponerlo “en movimiento para hacer conexiones y predicciones”, darle forma “para crear nuevos productos y resultados creativos”, como señala David Perkins<sup>5</sup>.

### Organización del aprendizaje en los Módulos obligatorios

Los módulos obligatorios organizan el aprendizaje en torno al desarrollo de una actividad de desempeño y actividades de evaluación que se integran. Los elementos que componen estos módulos son:

- **Visión panorámica del Módulo**

La visión panorámica de cada módulo presenta la gran idea, los objetivos de aprendizaje y conocimientos esenciales que se necesitan desarrollar para cumplir con el propósito formativo del módulo. Por último, se identifica el tiempo semanal y horas de clase propuestas para abarcar su implementación.

- **Propósito del módulo**

El propósito del módulo responde a tres interrogantes: ¿Qué se espera que los estudiantes comprendan?, ¿cómo se evidenciará que los estudiantes han comprendido? y ¿cómo tributa el módulo al marco formativo de las habilidades y actitudes del siglo XXI? Para responder a la primera interrogante se explica brevemente la gran idea que se pretende construir en el módulo. Luego, se relacionan explicativamente las habilidades, actitudes y conocimientos esenciales que pondrá en uso el estudiante para finalmente detallar cómo estos se integran y tributan al marco de habilidades y actitudes del s. XXI.

- **Ruta de aprendizaje**

Secuencia de 4 actividades de desempeño que describen sintéticamente qué habilidades –procedimientos estratégicos- y actitudes desarrollará el estudiante para poner en uso los conocimientos esenciales declarados en el módulo. Cada desempeño se construye identificando qué hace el estudiante – habilidad o procedimiento aplicado- y el conocimiento esencial que se moviliza. El conjunto de actividades de desempeño se integra coherentemente para dar cuenta del propósito formativo general declarado en el módulo.

---

<sup>5</sup> Perkins, D. Prólogo a Swartz, R. et al. (2017). Op. Cit., pág. 8.

- **Actividades de desempeño**

Para organizar el desarrollo de las actividades propuestas se utilizan criterios didácticos transversales que guíen flexiblemente a los docentes, de modo que puedan transferir la propuesta a los diferentes contextos. Los criterios utilizados se distinguen por su función didáctica, es decir, la finalidad formativa que se persigue a través de ello:

- Situación experiencial, permite enmarcar de forma situada un determinado aprendizaje, activando y enganchando el conocimiento previo con el nuevo conocimiento para desarrollar un aprendizaje significativo.
- Construcción del conocimiento, permite ilustrar cómo mediar, a través de una propuesta de selección de recursos y estrategias la adquisición y organización de nuevos conocimientos.
- Práctica guiada, modela paso a paso la mediación que realiza el docente, a través de actividades individuales, plenarias o colaborativas que desarrollan los estudiantes, para profundizar en la comprensión de un determinado conocimiento.
- Práctica independiente, detalla las actividades individuales y/o colaborativas que desarrollan los estudiantes para realizar desempeños flexibles que permitan profundizar y evidenciar su comprensión. Permite al docente monitorear el proceso de aprendizaje.
- Integración, corresponde a una actividad de síntesis que realiza el estudiante individualmente para evidenciar la comprensión del propósito declarado para la actividad. Por ejemplo, mediante el uso de ticket de salida.
- Orientaciones al docente: en esta sección se aclaran y precisan conceptos disciplinares que se han movilizado a la largo del módulo. Se realizan sugerencias complementarias al docente sobre el trabajo con adultos y/o estrategias didácticas que puedan facilitar su labor. Se sugieren seleccionar estrategias para guiar la retroalimentación y la evaluación formativa compartiendo criterios, estrategias de retroalimentación y rúbricas.

## MÓDULOS ELECTIVOS

Los módulos electivos ofrecen oportunidades de profundizar en el desarrollo de las habilidades y actitudes de los Objetivos de Aprendizaje del nivel y en la comprensión de las grandes ideas. Se desarrollan por medio de metodologías de aprendizaje basado en proyectos y aprendizaje basado en resolución de problemas, es decir, se organizan en torno a un tema que es planteado como problema o desafío y que permite ampliar el conocimiento esencial, profundizar en la comprensión de las grandes ideas y conectar con los intereses y experiencias de los estudiantes.

Los problemas y desafíos podrán ser adaptados a los contextos, intereses y experiencias vitales de los estudiantes.

Se sugiere considerar los Objetivos de Desarrollo Sostenible de la Agenda para el Desarrollo Sostenible de la UNESCO como foco para orientar los problemas y proyectos a desarrollar en los módulos electivos. Estos temas son<sup>6</sup>:

1. Fin de la pobreza
2. Hambre cero
3. Salud y bienestar
4. Educación de calidad<sup>7</sup>
5. Igualdad de género
6. Agua limpia y saneamiento
7. Energía asequible y no contaminante
8. Trabajo decente y crecimiento económico
9. Industria, innovación e infraestructura
10. Reducción de las desigualdades
11. Ciudades y comunidades sostenibles
12. Producción y consumo responsables
13. Acción por el clima
14. Vida submarina
15. Vida de ecosistemas terrestres
16. Paz, justicias e instituciones sólidas
17. Alianzas para lograr los objetivos.

Esta metodología debe permitir generar un compromiso activo del estudiante con el aprendizaje, lo cual se logrará si es que este aprendizaje: conecta con sus necesidades o inquietudes, y sabe de antemano cuál será este aprendizaje (*qué* aprender), lo considera importante (*por qué* aprender) y sabe *cómo* ocurrirá este aprendizaje (plan de trabajo) e idealmente participa en su planeamiento.

---

<sup>6</sup> Recuperado de: <https://es.unesco.org/sdgs>

<sup>7</sup> Las Bases Curriculares de EPJA se encuentran alineadas con este Objetivo N°4, en tanto apuntan al Aprendizaje a lo largo de la vida, y a una educación de calidad para todos.

## Estructura del aprendizaje en los módulos electivos

En coherencia con lo que plantean las Bases Curriculares, los módulos electivos ofrecen oportunidades para el desarrollo de metodologías de trabajo colaborativo y que aborden desafíos cognitivos y del entorno. En particular, los Programas de estudio desarrollan las metodologías de Aprendizaje Basado en Proyectos y Aprendizaje Basado en la Resolución de Problemas como propuestas que permiten desarrollar habilidades y poner en uso el conocimiento, integrar aprendizajes y promover la curiosidad y la búsqueda activa y creativa de respuestas. Estas metodologías buscan que los estudiantes puedan transferir el conocimiento a distintas áreas y/o situaciones de la vida real, por medio de aprendizajes significativos y relevantes. En cada nivel se ilustran dos ejemplos, uno de ABP y otro de Resolución de problemas, que podrán servir de modelo para que los docentes puedan construir nuevos proyectos o problemas.

Tanto en la Resolución de problemas como en ABP se busca conectar los problemas y preguntas con los Objetivos de Desarrollo Sostenible (ODS) de UNESCO, para reforzar su relevancia y transversalidad.

## Aprendizaje Basado en Proyectos

Consiste en la organización de los estudiantes en torno a una pregunta o desafío originado a partir de un problema real o que sea significativo para los estudiantes, que puede ser concreto o abstracto. En la medida que el problema es más complejo moviliza e integra diferentes áreas de conocimiento, promoviendo de esta manera la interdisciplinariedad. Para su desarrollo, es deseable que los docentes se organicen y planifiquen el trabajo de manera conjunta entre docentes de diferentes asignaturas.

Existe una serie de elementos que son requisitos para que el diseño de un proyecto permita maximizar el aprendizaje y la participación de los estudiantes, de manera que aprendan cómo aplicar el conocimiento al mundo real, cómo utilizarlo para resolver problemas, responder preguntas complejas y crear productos de alta calidad<sup>8</sup>. Estos elementos son:

- **Conocimiento esencial, comprensión y habilidades:**

El proyecto se enfoca en profundizar en la comprensión del conocimiento, ya que permite desarrollar a la vez los Objetivos de Aprendizaje y las habilidades del Siglo XXI que se requieren para realizar el proyecto.

Se basa en un problema significativo para resolver o una pregunta para responder, en el nivel adecuado de desafío para los alumnos, que se implementa mediante una pregunta de conducción abierta y atractiva.

---

<sup>8</sup> Adaptación de: John Larmer, John Mergendoller, Suzie Boss (ASCD 2015). *Setting the Standard for Project Based Learning: A Proven Approach to Rigorous Classroom Instruction*.

- **Indagación sostenida:**

El proyecto implica un proceso activo y profundo a lo largo del tiempo, en el que los estudiantes generan preguntas, encuentran y utilizan recursos, hacen preguntas adicionales y desarrollan sus propias respuestas.

- **Autenticidad:**

El proyecto tiene un contexto del mundo real, utiliza procesos, herramientas y estándares de calidad del mundo real y tiene un impacto real, ya que creará algo que será utilizado o experimentado por otros, y/o está conectado a las propias preocupaciones, intereses e identidades de los estudiantes.

Es importante saber en qué contexto del mundo real puede encontrarse el problema como el planteado y por qué el proyecto puede ser significativo para los estudiantes.

- **Voz y elección del estudiante:**

El proyecto permite a los estudiantes tomar algunas decisiones sobre los productos que crean, cómo funcionan y cómo usan su tiempo, guiados por el docente.

- **Gestión y auto organización:**

El proyecto exige a los estudiantes desarrollar el trabajo en equipo, la comunicación y la resolución de problemas; tomar decisiones sobre el diseño y la implementación del proyecto en sus distintas etapas. Esto implica identificar las competencias y procedimientos que son necesarios para desarrollar un plan de trabajo adecuado al proyecto, y una exploración activa de los recursos y actividades con que cuentan para su desarrollo. Asimismo, reconocer las fortalezas y debilidades con que cuenta cada uno de los miembros para su desarrollo.

- **Evaluación y Retroalimentación:**

El proyecto brinda oportunidades para que los estudiantes reflexionen sobre qué y cómo están aprendiendo. Incluye procesos de evaluación formativa y retroalimentación para que los estudiantes den y reciban comentarios sobre su trabajo, con el fin de revisar sus ideas y productos o realizar una investigación adicional.

- **Producto público.**

El proyecto requiere que los alumnos demuestren lo que aprenden, creando un producto que se presenta u ofrece a personas que se encuentran más allá del aula.

Considerando estos elementos, los programas proponen un diseño de ABP con la siguiente estructura:

Estructura	Descripción
<b>Problema central:</b>	Se describe el problema que origina el proyecto.
<b>Propósito:</b>	Refiere al propósito formativo del proyecto, es decir, qué se espera que aprendan los estudiantes gracias a la realización de este.
<b>Objetivos de Aprendizaje:</b>	Identifica y/o registra qué objetivos de aprendizaje de la asignatura y de otras asignaturas del nivel del plan de estudio de EPJA se integran para el desarrollo del proyecto.
<b>Preguntas:</b>	Se proponen preguntas orientadoras, que servirán para diseñar las etapas del proyecto. Son preguntas centrales y generales.
<b>Tipo de proyecto:</b>	Identifica el tipo de proyecto de acuerdo a las asignaturas que participan: STEM, interdisciplinario, etc.
<b>Producto:</b>	identifica el producto que se espera construir colaborativamente en el proyecto para dar respuesta concreta al problema.
<b>Habilidades y actitudes del siglo XXI:</b>	Identifica cuáles son las habilidades y actitudes que se desarrollarán, y a qué ámbito pertenecen.
<b>Etapas:</b>	se realiza un cronograma con las distintas etapas del proyecto, identificando: características de cada etapa, qué hará el estudiante, cómo lo realizará. Se apoya con recursos y/o ilustraciones cada etapa.
<b>Evaluación:</b>	Se comparten criterios de evaluación y rúbricas que guíen y permitan monitorear el desarrollo de los aprendizajes durante la realización del proyecto. Los criterios y las rúbricas deben verificar los aprendizajes de los objetivos que se identificaron para el proyecto, de manera descriptiva y por nivel de logro.
<b>Difusión final:</b>	describir cómo se difundirá el producto, incluyendo a la comunidad escolar y/o local.
<b>Recursos:</b>	Nombra recursos, clasificándolos según su tipo.

## Aprendizaje Basado en la Resolución de Problemas

El modelo de Aprendizaje basado en la Resolución de Problemas que presentan los Programas de estudio se organiza en torno a un problema o desafío cognitivo para el cual se busca encontrar una solución, por medio del uso del conocimiento y el desarrollo de habilidades. En los Programas, un problema se define por una situación o pregunta que presenta restricciones y cuya respuesta no es evidente.

Al resolver problemas, los estudiantes utilizan procesos y estrategias relacionadas con el análisis crítico, la investigación, la evaluación y la comunicación; planifican su trabajo y reflexionan sobre la solución que mejor responde a las restricciones que presenta el problema o desafío cognitivo. Como resultado, ponen en uso el conocimiento, lo amplían adquiriendo nuevos conceptos, principios e información, y desarrollan nuevas destrezas de pensamiento crítico y creativo<sup>9</sup>.

La resolución de problemas permite motivar y despertar el interés del estudiante, desarrolla la autonomía y el trabajo en equipo; esto requiere que las situaciones o problemas sean significativos y relevantes, y que puedan visualizar las posibles soluciones. Esta metodología requiere que el docente adquiera un rol activo como guía para monitorear el desarrollo del proceso y orientar el trabajo de los estudiantes.

---

<sup>9</sup> R. Swartz “El Aprendizaje basado en el Pensamiento. Cómo desarrollar en los alumnos las competencias del SXXI (2017). Edit. SM Figura 7-11. Pág. 232. Adaptación.



El modelo que proponen los Programas de estudio para el desarrollo del Aprendizaje Basado en la Resolución de Problemas se compone de los siguientes elementos:

Elementos de la estructura	Descripción
<b>Título</b>	Se plantea como una afirmación o pregunta que sintetiza el problema o desafío.
<b>Propósito</b>	Busca despertar el interés, predisponer al estudiante para el aprendizaje basado en problemas.
<b>Preparación</b>	Busca contextualizar a los estudiantes en la situación que se planteará y/o familiarizarlos con la resolución de problemas y su sistema de trabajo.
<b>Presentación del problema</b>	Se expone el problema, considerando la contextualización del mismo en una situación significativa. Se define con claridad y precisión cuál es el problema; se distinguen conceptos centrales y restricciones que constituyen el problema.
<b>Posibles soluciones</b>	Se describe cómo se mediará estratégicamente el trabajo colaborativo: el uso de estrategias para mediar disposiciones actitudinales positivas que les permitan a los estudiantes involucrarse con el problema y buscar soluciones (por ejemplo, la perseverancia), y estrategias de mediación para compartir las soluciones; se ilustran soluciones posibles que puede tener el problema.
<b>Investigación</b>	Describe cómo mediar el trabajo de investigación y el desarrollo de habilidades de indagación y evaluación; se ilustran recursos que se puedan utilizar y conocimientos disciplinares que se movilizan en la solución del problema.
<b>Evaluar la solución del problema:</b>	Describe cómo mediar estratégicamente las soluciones propuestas al problema, considerando las habilidades y la evaluación de las posibles soluciones.
<b>Comunicación</b>	Describe cómo se mediará la comunicación individual y/o colaborativa del problema, según códigos de comunicación pertinentes y característicos de las disciplinas.

## Orientaciones para evaluar los aprendizajes

La evaluación, como un aspecto intrínseco del proceso de enseñanza-aprendizaje, se plantea en estos programas con un foco formativo al servicio del aprendizaje de los estudiantes. Para que esto ocurra, se plantea recoger evidencias que permitan describir con precisión la diversidad existente en el aula para tomar decisiones pedagógicas y retroalimentar a los estudiantes. La evaluación desarrollada con foco pedagógico favorece la motivación de los estudiantes a seguir aprendiendo; asimismo, el desarrollo de la autonomía y la autorregulación potencia la reflexión de los docentes sobre su práctica y facilita la toma de decisiones pedagógicas pertinentes y oportunas que permitan apoyar de mejor manera los aprendizajes.

Para implementar una evaluación con un foco formativo, se requiere:

- Diseñar experiencias de evaluación que ayuden a los estudiantes a poner en práctica lo aprendido en situaciones que muestren la relevancia o utilidad de ese aprendizaje.
- Evaluar solamente aquello que los alumnos efectivamente han tenido la oportunidad de aprender mediante las experiencias de aprendizaje mediadas por el profesor.
- Procurar que se utilice diversas formas de evaluar, que consideren las distintas características, ritmos y formas de aprender, necesidades e intereses de los estudiantes, evitando posibles sesgos y problemas de accesibilidad para ellos.
- Promover que los alumnos tengan una activa participación en los procesos de evaluación; por ejemplo: al elegir temas sobre los cuales les interese realizar una actividad de evaluación o sugerir la forma en que presentarán a otros un producto; participar en proponer los criterios de evaluación; generar experiencias de auto y coevaluación que les permitan desarrollar su capacidad para reflexionar sobre sus procesos, progresos y logros de aprendizaje.
- Que las evaluaciones sean de la más alta calidad posible; es decir, deben representar de la forma más precisa posible los aprendizajes que se busca evaluar. Además, las evidencias que se levantan y fundamentan las interpretaciones respecto de los procesos, progresos o logros de aprendizajes de los estudiantes, deben ser suficientes como para sostener de forma consistente esas interpretaciones evaluativas.

El profesor puede utilizar diferentes métodos para evaluar los OA. Para esto, se sugiere emplear una variedad de medios y evidencias, como portafolios, registros anecdóticos, proyectos de investigación grupales e individuales, informes, presentaciones, entre otros. La forma en que se diseñe este tipo de evaluaciones y el modo en que se registre y comunique la información que se obtiene de ellas debe permitir que dichas evaluaciones integren lo formativo y sumativo para retroalimentar tanto la enseñanza como el aprendizaje.

El uso formativo de la evaluación debiera preponderar en las salas de clases, utilizándose de manera sistemática para reflexionar sobre el aprendizaje y la enseñanza, y para tomar decisiones pedagógicas pertinentes y oportunas que busquen promover el progreso del aprendizaje de todos los estudiantes, considerando la diversidad como un aspecto inherente a todas las aulas.

El proceso de evaluación formativa que se propone implica articular el proceso de enseñanza-aprendizaje en función de responder a las siguientes preguntas: **¿A dónde voy?** (qué objetivo de aprendizaje espero lograr),

¿Dónde estoy ahora? (cuán cerca o lejos me encuentro de lograr ese aprendizaje) y ¿Qué estrategia o estrategias pueden ayudarme a llegar a donde tengo que ir? (qué pasos tengo que dar para acercarme a ese aprendizaje). Este proceso continuo de establecer un objetivo de aprendizaje, evaluar los niveles actuales y luego trabajar estratégicamente para reducir la distancia entre los dos, es la esencia de la evaluación formativa. Una vez que se alcanza una meta de aprendizaje, se establece una nueva meta y el proceso continúa.

Para promover la motivación para aprender, el nivel de desafío y el nivel de apoyo deben ser los adecuados –en términos de Vygotsky (1978), estar en la zona de desarrollo próximo de los estudiantes–, para lo cual se requiere que todas las decisiones que tomen los profesores y los propios estudiantes se basen en la información o evidencia sobre el aprendizaje recogidas continuamente<sup>10</sup>.

Como parte de la evaluación formativa, los Programas proponen en cada actividad un conjunto de criterios que permiten evaluar el desempeño de los estudiantes en un determinado aprendizaje. Estos criterios permiten identificar el lugar en que se encuentran los estudiantes en el desarrollo de las habilidades y la construcción de conocimientos, entregando información que permita al docente tomar decisiones pedagógicas para avanzar hacia el logro de los aprendizajes propuestos<sup>11</sup>.

Los criterios de evaluación describen el dominio de conceptos, de procedimientos y actitudes en los estudiantes. En su conjunto, permiten evaluar la comprensión y la disposición o inclinación a actuar de acuerdo al marco de Habilidades y actitudes del siglo XXI. Cuando se integran en el desarrollo de la clase, los criterios de evaluación permiten generar un mejoramiento continuo del aprendizaje<sup>12</sup>.

---

<sup>10</sup> Mineduc (2017). *Presentación de Criterios de evaluación, calificación y promoción al Consejo Nacional de Educación*. Fundamentos a la propuesta de actualización de criterios y normas de Evaluación, Calificación y Promoción Escolar de estudiantes de Educación Regular presentada por la Unidad de Currículo y Evaluación al Consejo Nacional de Educación. Santiago, pág. 74.

<sup>11</sup> Para la construcción de los criterios, se han tenido a la vista las orientaciones que plantea la Mesa Covid Universitaria y la normativa vigente para la atención a la diversidad, la inclusión y la flexibilidad en la repuesta educativa contenida en la ley 20.845 de inclusión escolar (Art. 1°, núm. i), y como referente los principios del Decretos 83 de 2015 y Decreto 67 de 2018.

<sup>12</sup> Propuestas Educación Mesa Social Covid-19 (2021). *Recomendación para una evaluación pertinente en tiempos de crisis*. Santiago de Chile. Santiago, pág. 65.

## Referencias

John Larmer, John Mergendoller, Suzie Boss. *Setting the Standard for Project Based Learning: A Proven Approach to Rigorous Classroom Instruction*, (ASCD 2015).

Knowles, M. S., Holton III, E. F., & Swanson, R. A. (2014). *The adult learner: The definitive classic in adult education and human resource development*. Routledge.

Lemov, D. (2014). *Teach like a champion 2.0: 62 techniques that put students on the path to college*. John Wiley & Sons.

Mineduc (2017). *Presentación de Criterios de evaluación, calificación y promoción al Consejo Nacional de Educación*. Fundamentos a la propuesta de actualización de criterios y normas de Evaluación, Calificación y Promoción Escolar de estudiantes de Educación Regular presentada por la Unidad de Currículo y Evaluación al Consejo Nacional de Educación. Santiago

Propuestas Educación Mesa Social Covid-19 (2021). *Recomendación para una evaluación pertinente en tiempos de crisis*. Santiago de Chile. Santiago, pág. 65

R. Swartz "El Aprendizaje basado en el Pensamiento. Cómo desarrollar en los alumnos las competencias del SXXI (2017). Edit. SM Figura 7-11. Pág. 232. Adaptación.

Universidad del Desarrollo, Centro de Innovación. Aprendizaje Basado en Problemas (ABP). Recurso web disponible en: <https://innovaciondocente.udd.cl/metodologias-activas/>

UNESCO (2015). *La Agenda para el Desarrollo Sostenible*.  
<https://www.un.org/sustainabledevelopment/es/development-agenda/>

# Presentación Pensamiento Computacional

## Propósito Formativo

El pensamiento computacional permite a los jóvenes y adultos adquirir habilidades y conocimientos para resolver problemas, diseñar sistemas y comprender el mundo en que vivimos desde los conceptos fundamentales de la informática (Wing, 2006), transformándose durante los años en una competencia imprescindible de la era digital y del siglo XXI.

En este sentido, resolver problemas apunta a que los estudiantes sean productores activos de conocimiento, al utilizar saberes y experiencias ya existentes para dar respuesta a una o más situaciones complejas, pero también enfatiza el descubrimiento al combinar reglas propias de la informática y aplicarlas a situaciones novedosas, auténticas y concretas para ellos. Además, por medio de la resolución de problemas desarrollan el pensamiento crítico al discriminar y evaluar distintos conocimientos de la informática, las tecnologías y sistemas de la información, y el ámbito digital, estableciendo condiciones para elaborar juicios informados y tomando decisiones como ciudadanos de un mundo en constante cambio y evolución.

Sobre este escenario, se promueve que los estudiantes diseñen sistemas de soluciones con impronta creativa, abordando las situaciones problemáticas desde distintos puntos de vista y reuniendo los datos necesarios y disponibles, considerando las restricciones del contexto y cumpliendo con criterios de eficiencia y efectividad. Siguiendo esta lógica, la flexibilidad y perseverancia son actitudes del siglo XXI que potencian la propuesta de soluciones para satisfacer necesidades y deseos de manera sostenible.

En consecuencia, las Bases Curriculares favorecen el aprendizaje de conceptos de programación y cómo aprovechar las posibilidades que los sistemas digitales y de información proveen, no solo desde la perspectiva de usuarios sino también desde el rol de creadores de tecnologías, con una mirada hacia las crecientes transformaciones sociales que las ciencias de la computación y la inteligencia artificial están produciendo. El pensamiento computacional se ha transformado progresivamente en la nueva alfabetización digital, por tanto, los conocimientos tecnológicos que poseen los estudiantes son la base para potenciar habilidades de resolución de problemas, pensamiento abstracto, pensamiento crítico, razonamiento lógico y analítico, creatividad, colaboración y comunicación.

Considerando lo anterior, las Bases Curriculares de Pensamiento Computacional del presente ciclo de Formación de Educación Media, buscan contribuir a la formación de Personas Jóvenes y Adultos con el dominio de habilidades que les permitan utilizar, expresar y desarrollar sus ideas por medio de la tecnología de la información y la comunicación, a un nivel adecuado para su futuro laboral y sus estudios superiores, y logren ser participantes activos en el espacio digital (Mineduc, 2019).

## Enfoque de la asignatura

La asignatura pone énfasis en el desarrollo de habilidades y formas de razonamiento propias de la informática, aplicadas a la formulación y resolución de problemas de forma flexible y proactiva para encontrar soluciones innovadoras.

Estas lógicas de razonamiento tienen su origen en las ciencias, pero principalmente, en el pensamiento ingenieril, pensamiento científico y pensamiento matemático, cuyas implicancias se asocian a una manera de pensar, de diseñar, de probar y usar modelos computacionales, así como a una manera específica de abordar problemas del mundo real mediante la descomposición, la evaluación, la abstracción, la generalización y el pensamiento algorítmico. Con esto, se busca conformar una estructura de pensamiento que integre los aprendizajes que los estudiantes han adquirido en la Educación Básica, entre los que se destacan modelos, funciones o ecuaciones matemáticas; el análisis, explicación y representación de las partes de la célula; los sistemas de órganos del ser humano; la clasificación de los seres vivos en diversos grupos taxonómicos; la evaluación de diversos componentes de un circuito eléctrico o el diseño de experimentos.

En este sentido, aprender pensamiento computacional contempla la aplicación de un razonamiento que permite organizar y analizar datos de forma lógica y formular problemas que puedan ser resueltos mediante dispositivos electrónicos u otras herramientas para trabajar en búsqueda de una solución. Así, también implica representar datos de manera abstracta con modelos y simulaciones, identificar y analizar posibles soluciones con el objeto de encontrar la combinación de pasos de manera eficiente y efectiva, implementar soluciones sobre la base de una serie de pasos ordenados, y generalizar y transferir ese proceso de solución a nuevos problemas.

Junto con esto, esta asignatura brinda un contexto para el estudio de la programación de gran alcance, con una aplicación que va más allá de la tecnología y la programación en sí misma, entendiendo que esta corresponde a una de las prácticas más visibles del pensamiento computacional. Esta perspectiva posibilita profundizar en el desarrollo de competencias necesarias para la vida laboral, como son las competencias digitales asociadas al manejo de la información, la colaboración en línea, el ejercicio de la ciudadanía digital, la creación de contenidos en línea y la resolución de problemas en y con internet.

Las experiencias cotidianas de los estudiantes de la educación para Personas Jóvenes y Adultas permiten situar y dar sentido a lo que sucede en un mundo fuertemente influenciado por las tecnologías. En este contexto, el desarrollo de las habilidades del pensamiento computacional contribuye a propiciar un rol activo, honesto y responsable de los estudiantes frente a la sociedad, en el que se releva la importancia de la dimensión ética y de reflexionar en torno a temas controversiales, considerando, entre otros: las brechas de género que existen en el acceso a la programación y al uso de tecnologías digitales; y los desafíos en términos de inclusión, gestión de la diversidad y oportunidades para personas con necesidades especiales que ofrece hoy la tecnología. De esta manera, el enfoque de la asignatura favorece una mejor comprensión de la realidad social y promueve una actitud activa en su mejoramiento.

## Organización curricular

Las Bases Curriculares de Pensamiento Computacional se articulan en torno a Objetivos de Aprendizaje de habilidades y actitudes, los cuales fortalecen disposiciones para un saber hacer que integre el respeto a los principios éticos y valores fundamentales de la sociedad. Asimismo, los Objetivos de Aprendizaje se entrelazan con conocimientos esenciales que buscan focalizar la comprensión de las ideas y procesos fundamentales de la asignatura. Las grandes ideas operan como propósito formativo del nivel 1 y 2, orientando la comprensión y la articulación de los Objetivos de Aprendizaje y los conocimientos esenciales.

## Objetivos de Aprendizaje

Los Objetivos de Aprendizaje de la asignatura se organizan en el nivel 1 y 2 para desarrollar las habilidades y lógicas de razonamiento del pensamiento computacional, fundamentadas en las grandes ideas de la informática y en el valor de la abstracción como forma de pensamiento diferenciadora. Los objetivos integran actitudes de los ámbitos que organizan las habilidades del siglo XXI de acuerdo con un criterio de pertinencia para ser trabajadas integralmente con las habilidades.

Para esta asignatura, se han organizado los Objetivos de Aprendizaje en<sup>13</sup>:

- Descomponer
- Evaluar
- Modelar
- Generalizar y transferir
- Pensar de forma algorítmica.

### Descomponer

Pensar en términos de descomponer facilita comprender las tecnologías artefactuales en términos de cada una de sus parte o componentes, lo que implica que cada una de las partes debe comprenderse, desarrollarse y evaluarse por separado.

En este eje se desarrollan las actitudes del ámbito Maneras de pensar. En Educación Media se focaliza en promover la actitud de pensar con apertura a distintas perspectivas y contextos, asumiendo riesgos y responsabilidades.

### Evaluar

Pensar en términos de evaluar significa desarrollar la habilidad de pensar para asegurar que una solución, ya sea un algoritmo, un sistema o un proceso, cumpla eficientemente con su propósito de acuerdo con el contexto donde se presenta la problemática.

En este eje se desarrollan las actitudes del ámbito Maneras de Trabajar. Así, en la Educación Media se promueve trabajar con honestidad en la realización de tareas colaborativas con miras al logro de una meta común.

### Modelar

<sup>13</sup> Basado en *Computational Thinking: A Guide for Teachers*, 2015.

Modelar es una habilidad que permite hacer más comprensible un sistema u objeto del mundo real; al utilizar el pensamiento abstracto, facilita comprender y estudiar con mayor eficiencia uno o más problemas al extraer los elementos que son más relevantes de una situación.

En este eje se desarrollan las actitudes del ámbito Herramientas para trabajar. Esto implica que en la Educación Media se favorece el aprovechar las herramientas disponibles para aprender y resolver problemas.

### Generalizar y Transferir

La generalización facilita identificar patrones, similitudes y conexiones con la intención de transferir las maneras de solucionar un problema a otras situaciones similares de manera eficaz y eficiente.

En este eje se desarrollan las actitudes de Maneras de vivir el mundo. Así, en Educación Media, se promueve que los estudiantes actúen con honestidad, responsabilizándose por las propias acciones y decisiones con conciencia de las implicancias que estas tienen sobre uno mismo y los demás.

### Pensar de forma algorítmica

El pensamiento algorítmico permite llegar a una solución a través de una identificación clara de pasos y reglas, cuando se está frente a problemas que tienen que ser resueltos de forma continua, de manera que no se tengan que pensar una y otra vez los mismos caminos para llegar a resolverlos.

En este eje se desarrollan las actitudes del ámbito de las Herramientas para Trabajar. En el caso de la Educación Media, implica que los estudiantes actúen de acuerdo con los principios de la ética en el uso de la información y de la tecnología, respetando la privacidad de los otros.

### OBJETIVOS DE APRENDIZAJE

Se espera que los estudiantes sean capaces de:

**OA1.** Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que los componen. **(Descomponer)**

**OA2.** Organizar datos de una situación problemática estableciendo secuencias e identificando fallas y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

**OA3.** Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

**OA4.** Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. **(Generalizar y Transferir)**

**OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y Transferir)**

**OA6.** Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

### Conocimientos esenciales

- Procesos y funcionamiento de las cosas para resolver problemas cotidianos.
- Modelos entidad relación y simulación de sistemas.



- Sistemas computacionales: dispositivos, *hardware* y *software*, y resolución de problemas.
- Redes e Internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.
- Algoritmos y programación: lenguajes, métodos y entornos de programación.
- Representación y análisis de datos.
- Impacto social de la computación: interacciones, cultura, seguridad y ética.

## Orientaciones para el docente

### Orientaciones didácticas:

Las orientaciones que se presentan a continuación destacan elementos que son relevantes al momento de implementar el programa. Estas orientaciones se vinculan estrechamente con el logro de los Objetivos de Aprendizaje especificados en las Bases Curriculares.

El enfoque de la educación en Pensamiento Computacional que orienta el programa se basa en el desarrollo de habilidades y formas de razonamiento propias de la informática, aplicadas a la formulación y resolución de problemas de forma flexible y proactiva, para encontrar soluciones innovadoras. También, contempla la aplicación de un razonamiento que permite organizar y analizar datos de forma lógica y formular problemas que puedan ser resueltos mediante dispositivos electrónicos u otras herramientas para trabajar en búsqueda de una solución. Asimismo, desarrolla habilidades necesarias para el pensamiento computacional, tales como Descomponer, Evaluar, Modelar Generalizar y Transferir y Pensar de forma Algorítmica. En consecuencia, al organizar el proceso de aprendizaje, el docente debe tomar en cuenta los siguientes factores:

- **Integración de los Objetivos de Aprendizaje:** Las actividades modélicas que se ofrecen en este programa consideran en numerosas oportunidades el desarrollo de los objetivos de aprendizaje de manera integrada. De igual manera, las actitudes se deben trabajar conjuntamente con los Objetivos de Aprendizaje.
- **Planificación:** Es necesario incluir un tiempo al inicio de una unidad para rescatar las experiencias y conocimientos previos de los estudiantes, pues constituyen el punto de partida para implementar adecuadamente las actividades. Asimismo, se debe considerar un enfoque participativo y colaborativo que implique que los estudiantes tengan el protagonismo en sus aprendizajes. Durante el desarrollo de las actividades, se debe promover una relación positiva con el aprendizaje en todos los alumnos. Para esto, es necesario que se consideren instancias de retroalimentación de los procesos de expresión, creación y apreciación, aprovechando descubrimientos, errores y dificultades como fuente de nuevos aprendizajes y logros. Para orientar y apoyar al estudiantado en la superación y solución de dificultades, es importante que conozcan y comprendan los criterios de evaluación al inicio de cada actividad. Al finalizar cada módulo, se debe considerar un tiempo adecuado para que expongan, justifiquen y enjuicien críticamente sus proyectos, algoritmos o programas lo que les permite observar sus progresos y valorar sus trabajos y los de sus pares.

- Desarrollo del razonamiento lógico:** el razonamiento lógico permite a los estudiantes dar sentido a los problemas mediante el análisis y comprobación de hechos para aprender a pensar de forma efectiva y clara. Esto implica que el profesor favorezca que los alumnos pongan en tensión sus propios conocimientos y modelos internos para predecir y sacar conclusiones respecto de una situación o más situaciones que requieran solución. El razonamiento lógico es sumamente relevante en la asignatura, este permite a los estudiantes depurar sus propios códigos para aislar errores e idear mejoras empleando la abstracción, la evaluación y el pensamiento algorítmico.
- Tecnologías:** el docente debe considerar que las herramientas tecnológicas computacionales actuales varían en su efectividad a través del tiempo, al permitir progresivamente mayor interacción con diferentes componentes artefactuales y también humanos, siendo imperativo para esta asignatura impulsar el desarrollo de nuevas herramientas diseñadas y elaboradas expresamente para fomentar el pensamiento computacional con miras a un uso contextualizado y cotidiano. Es necesario que los estudiantes estén conscientes de que los computadores no funcionan a menos que un programa se esté ejecutando y que un amplio rango de dispositivos pueden ser considerados un computador, por lo tanto, es importante establecer la relación entre las personas, la tecnología, los usos y que diferentes tecnologías responden a diferentes o iguales necesidades.

Dentro de este uso y la comunicación que existe entre usuarios, computadores, programadores y empresas, la ciberseguridad o seguridad de la información en el mundo digital es compleja en el sentido de los conceptos de la invasión de la privacidad, aumento de la vigilancia de datos y de las transacciones comerciales y cívicas digitales<sup>14</sup>. Esto tiene una connotación de responsabilidad para los estudiantes en su rol de programadores y usuarios de programas. En esta asignatura y a nivel instrumental, se sugiere la lectura de las políticas de privacidad y protección de datos personales. Teniendo presente el alcance del uso de los datos y del tipo de datos personales a los cuales se pueden acceder preguntando a los usuarios<sup>15</sup>. En cuanto a los consejos de autocuidado, los estudiantes deben tener claro que las personas son las dueñas de los datos personales y no quienes las administran, lo cual significa que hay derechos y deberes sobre el uso y los fines que se les da a los datos<sup>16</sup>.
- Aprendizaje colaborativo:** En la elaboración de las múltiples tareas de la asignatura, es importante que el docente favorezca la comunicación y la colaboración entre los estudiantes, tanto al momento de programar, diseñar o ejecutar un programa. Esto permite profundizar y compartir la construcción de aprendizajes, junto con potenciar habilidades y actitudes transversales como el respeto por la opinión de los pares y el descubrimiento de la riqueza presente en la diversidad.

---

<sup>14</sup> Pangrazio, L. & Sefton-Green, J. (2021). Digital Rights, digital Citizenship and Digital Literacy: What's the Difference? *Journal of new approach in education research*, 10(1). 15-27

<sup>15</sup> Servicio Nacional del Consumidor. <https://www.curriculumnacional/link/> <https://www.sernac.cl/portal/617/w3-article-53061.html>

<sup>16</sup> Servicio Nacional del Consumidor. <https://www.curriculumnacional/link/> <https://www.sernac.cl/portal/607/w3-article-3389.html>

- **Ciudadanía y alfabetización digitales:** Con la influencia cada vez más creciente de las tecnologías en la sociedad, es sumamente relevante que el docente favorezca la comprensión del funcionamiento y los principios que rigen el entorno digital por parte de los estudiantes. Asimismo, es necesario que en todo momento se pueda realizar un análisis del rol que juegan las tecnologías en la sociedad, la incidencia en la vida cotidiana propia y la de los demás, así como el papel que ellas ocupan en la construcción del conocimiento. Es necesario que la tecnología sea vista como sujeta de valoración en cuanto a su uso y creación, siendo imprescindible la reflexión crítica y ética sobre la misma.

Ciudadanía digital se entiende como el derecho a participar en la sociedad en línea, la cual considera el acceso, inclusión y libertades comunicativas. Una vez que se tiene acceso, el estudiante debe focalizar su desarrollo como ciudadano digital considerando algunas condiciones, como descifrado de las fuentes de las noticias o bien construcción de las identidades digitales<sup>17</sup>. El estudiante debe tomar en cuenta las fuentes, la interpretación de la información, los flujos locales y globales, y sobre todo anticipar posibles consecuencias de lo expuesto como ciudadano digital.

En este sentido, la alfabetización digital es fundamental para lograr un ciudadano digital y se puede entender como una educación digital que permite al estudiante comprender la información, las normas y las convenciones que rodean a lo digital y las formas para demostrar su adquisición. Esta asignatura tiene foco en la alfabetización digital utilizando el concepto de aprender programando y la educación digital con sus normas y convenciones son utilizadas para lograr un programa que pueda ser utilizado por un ciudadano digital<sup>18</sup>.

### Orientaciones para la evaluación en Pensamiento Computacional

La evaluación ayuda tanto al docente como al estudiante a determinar las fortalezas y debilidades que surgen durante el proceso de enseñanza y aprendizaje. Con esta información, se puede tomar decisiones para modificar la planificación y adecuarla a las características y necesidades de los alumnos. Por su parte, ellos podrán focalizar sus esfuerzos con la confianza de mejorar sus resultados a partir de la orientación que la evaluación proporciona.

En Pensamiento Computacional, como en todas las asignaturas, es muy importante contar con criterios claros y conocidos por los estudiantes, lo que ayuda a transparentar la evaluación. En este programa, se proporcionan ejemplos de criterios y pautas de evaluación, los que se recomienda contextualizar según la situación de aprendizaje específica.

Para poder evaluar los Objetivos de Aprendizaje, se debe considerar que los mismos tributan a habilidades y actitudes que buscan configurar de forma dinámica el desarrollo del pensamiento computacional por medio de propuestas de aprendizaje que refieren a procedimientos individuales y en equipos, los cuales son necesarios de monitorear, acompañar y considerar en una lógica de proceso sistemática y rigurosa. Sobre este contexto, se deben favorecer prácticas de evaluación que consideren:

1. Colaborar en torno a la informática.
2. Reconocer y definir problemas computacionales.

<sup>17</sup> Pangrazio, L. & Sefton-Green, J. (2021). Digital Rights, digital Citizenship and Digital Literacy: What's the Difference? *Journal of new approach in education research*, 10(1). 15-27

<sup>18</sup> Ibidem.

3. Desarrollar y usar abstracciones.
4. La creación de artefactos computacionales.
5. Probar la tecnología disponible.
6. Comunicarse por medio de tecnologías.

En este sentido, para que la evaluación contribuya efectivamente a lograr los aprendizajes, se deben aplicar gradualmente los criterios o pautas de evaluación, tanto durante el desarrollo del proceso como en la presentación de un producto final. En estricto rigor, todas las actividades que se realizan deben ser evaluadas; solo así el docente puede asegurar su pertinencia para el logro de un OA. Estas evaluaciones no deben traducirse necesariamente en una calificación; su fin es proporcionar evidencia acerca del grado de avance en el logro del o los Objetivos de Aprendizaje y apoyar o reorientar el trabajo de los alumnos en caso de ser necesario.

Para lograr un proceso de evaluación efectivo es necesario realizar un diseño previo, para lo cual se sugiere responder a algunas preguntas como:

- ¿Cuáles son los Objetivos de Aprendizaje y/o metas que abarcará la evaluación?
- ¿Qué criterios son adecuados a las actividades efectivamente realizadas y los propósitos de la evaluación?
- ¿Qué evidencias en el trabajo de los estudiantes permiten demostrar los logros esperados?
- ¿Cuáles son los criterios para definir la calidad en un logro de aprendizaje?
- ¿Qué estrategias y procedimientos son los adecuados?
- ¿Quién o quiénes realizarán la evaluación?
- ¿En qué momentos se hará?
- ¿Cómo se comunicarán los resultados?

### Procedimientos de evaluación

Los procedimientos son los métodos o situaciones que se consideran útiles para recoger evidencia en relación con el avance de los aprendizajes. A continuación, se sugieren algunos procedimientos que diversifican y actualizan la forma en que se puede desarrollar este proceso y evaluar el logro de los OA en la asignatura, tanto en instancias formativas como con fines sumativos.

**Portafolio:** Permite evaluar el rendimiento de los estudiantes de forma auténtica en el aula, pues las tareas y actividades de aprendizaje pueden flexibilizarse en busca de una solución o respuesta a un problema o pregunta a resolver, permitiéndole a los alumnos demostrar sus aprendizajes de diferentes formas, pues así se facilita evidenciar la creatividad, la motivación, el razonamiento y las capacidades que van desarrollando progresivamente. El portafolio, concretamente, puede ser una carpeta tradicional o una carpeta digital, donde se guardan y organizan trabajos prácticos, actividades de clase o elementos de un proyecto, a los cuales se puede acceder fácilmente, siendo evidencia para el profesor y, a la vez, una forma de autoevaluación por parte del estudiante. El portafolio es útil para medir el uso de algoritmos, las abstracciones, el modelamiento y la resolución de problemas, que generalmente son difíciles de evaluar por medio de preguntas abiertas u opciones múltiples.

**Proyectos:** Consiste en plantear un problema o situación que requiera una solución informática para resolver problemas sencillos o complejos o efectuar una investigación guiada o modelar un problema real. Es necesario fijar con precisión los objetivos, considerar resultados abiertos, identificar los recursos necesarios, favorecer el trabajo en equipo y conectar de forma ideal diferentes áreas del conocimiento. Se recomienda que para cada fase del proyecto se tenga una pauta de evaluación que oriente la recolección de evidencia de aprendizajes durante el proceso de aprendizaje, especialmente cuando los alumnos desarrollen algoritmos.

**Desarrollo de programas:** Permite evaluar el proceso de diseño de un programa que frecuentemente se repite hasta que el estudiante haya dado con la solución necesaria. Con este procedimiento, los estudiantes aprenden por qué las personas desarrollan programas, las ventajas y desventajas del diseño de un algoritmo, y cómo las decisiones del programador involucran restricciones, eficiencia, ética y pruebas de usuarios. Como todo procedimiento, debe realizarse usando una pauta de evaluación basada en criterios compartidos previamente con los alumnos. Es importante considerar que el diseño del programa, así como los algoritmos necesarios y lenguaje propiamente tal, pueden ser desarrollados usando un computador o similar, o bien recursos análogos.

Una recomendación importante es que, al construir una pauta de evaluación, ya sea basada en ítems de valoración o rúbricas, es necesario tener presente que las diferentes dimensiones de un desempeño (proyecto, código, modelos u otro) se ponderen de acuerdo a su relevancia en el OA; esto significa, por ejemplo, que si el OA pone énfasis en la descomposición, no puede asignarse a este aspecto el mismo peso (puntaje máximo) que a aspectos menos relevantes en dicho OA, como podría ser la destreza en la ejecución de procedimiento (ingresar datos).

### Orientaciones para la contextualización

El programa propone cuatro módulos alineados con los Objetivos de Aprendizaje para los que se consideran actividades modélicas. Los docentes pueden contextualizar y adaptar esta propuesta a su realidad educativa, pues los módulos brindan la oportunidad de ser desarrollados aprovechando las herramientas tecnológicas disponibles, como equipos informáticos, o bien como recursos análogos. Dado este contexto, cada módulo tiene una orientación hacia el uso de las tecnologías digitales disponibles, sin embargo, cada actividad puede ser resuelta utilizando materiales concretos, como bloques, cubos o lápiz y papel. Sobre este contexto es necesario que las actividades siempre estén en directa relación con los Objetivos de Aprendizaje.

Dichas actividades promueven el pensamiento algorítmico, el análisis de datos, la creatividad y la resolución de problemas considerando las características, intereses y contextos de los estudiantes, así como también la infraestructura y recursos disponibles en el establecimiento educacional. Esto significa seleccionar recursos, plataformas, juegos, aplicaciones o simulaciones dentro de un entorno significativo para los estudiantes, que permita a su vez recopilar datos, analizar los logros y comunicar resultados

Es importante que cada profesor sea capaz de adaptar los diferentes elementos de su planificación (por ejemplo: tiempos o recursos), cuando en su curso tenga estudiantes que presenten algún tipo de discapacidad física, psicológica, cognitiva u otras. Esto es particularmente necesario en el caso de estudiantes con capacidades visuales, auditivas o de movilidad limitadas o diferentes. El docente podrá buscar estrategias y adaptaciones pedagógicas para desarrollar las habilidades propias de la asignatura, pudiendo usar recursos concretos o digitales para modelar problemas, desarrollar códigos, analizar datos o evaluar el impacto de la tecnología en la sociedad.

En cuanto a la diversidad cultural, al planificar e implementar las actividades es necesario tener presentes las culturas de origen del curso, poniendo atención a los problemas que se plantean y a aspectos relativos con las brechas digitales que puedan existir en cuanto a conectividad o acceso a equipos informáticos. En ambos casos, se debe procurar la participación y apoyo de todos, lo que los preparará para comprender, aceptar y valorar la diversidad.

## Visión panorámica de módulos del nivel

### Módulos obligatorios

#### M1 Nivel 1 y 2 EM

¿Cómo descomponer problemas para evaluarlos usando diferentes fuentes de información, modelos y herramientas?

#### M2 Nivel 1 y 2 EM

¿Cómo se modelan los niveles de interacción entre los sistemas computacionales y los sistemas sociales, ambientales o industriales que enfrenta un usuario?

#### M3 Nivel 1 y 2 EM

¿Cómo transferir la tipología de sistemas y redes para solucionar problemas de seguridad, acceso y control de datos?

#### M4 Nivel 1 y 2 EM

¿Por qué las personas evalúan y seleccionan algoritmos en función del rendimiento, reutilización y la facilidad de implementación para resolver un problema?

### Módulos electivos

#### ME1 Nivel 1 y 2 EM

¿Por qué los avances de la computación cambian las actividades cotidianas de las personas?

#### ME2 Nivel 1 y 2 EM

¿Cómo crear un programa computacional que genere un impacto amplio a nivel social?

#### ME3 Nivel 1 y 2 EM

¿Por qué la sociedad y los ámbitos de la actividad humana se ven afectados por los grados de comunicación que ofrece la informática?

#### ME4 Nivel 1 y 2 EM

¿Cómo el uso de la tecnología impacta en las oportunidades de las personas en la sociedad?

## Visión panorámica de Objetivos de Aprendizaje y conocimientos esenciales de Pensamiento Computacional

Módulos obligatorios	Módulo 1 Nivel 1 o 2 EM	Módulo 2 Nivel 1 o 2 EM	Módulo 3 Nivel 1 o 2 EM	Módulo 4 Nivel 1 o 2 EM
Gran idea módulo	Los problemas pueden descomponerse en partes más pequeñas y manejables, organizarse siguiendo patrones, y solucionarse en base a reglas, modelos y algoritmos.	El comportamiento de un sistema computacional (dispositivos, <i>hardware</i> y <i>software</i> ) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema.	El comportamiento de un sistema computacional (dispositivos, <i>hardware</i> y <i>software</i> ) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema	El pensamiento algorítmico permite organizar pasos y reglas para diseñar soluciones y resolver una diversidad de problemas cuya comprensión se favorece al integrar conceptos matemáticos como variables, funciones, ecuaciones o gráficos, entre otros.
Objetivos de Aprendizaje	<b>OA1.</b> Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que lo componen. (Descomponer)	<b>OA2.</b> Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. (Evaluar)	<b>OA4.</b> Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. (Generalizar y Transferir)	<b>OA4.</b> Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada.  (Generalizar y Transferir)



Módulos obligatorios	Módulo 1 Nivel 1 o 2 EM	Módulo 2 Nivel 1 o 2 EM	Módulo 3 Nivel 1 o 2 EM	Módulo 4 Nivel 1 o 2 EM
	<b>OA2.</b> Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. <b>(Evaluar)</b>	<b>OA3.</b> Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. <b>(Modelar)</b>	<b>OA5.</b> Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. <b>(Generalizar y Transferir)</b>	<b>OA5.</b> Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. <b>(Generalizar y Transferir)</b>
	<b>OA3.</b> Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. <b>(Modelar)</b>		<b>OA6.</b> Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. <b>(Pensar de forma algorítmica)</b>	<b>OA6.</b> Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. <b>(Pensar de forma algorítmica)</b>
Conocimientos esenciales	<ul style="list-style-type: none"> <li>• Procesos y funcionamiento de las cosas para resolver problemas cotidianos.</li> <li>• Modelos entidad relación y simulación de sistemas.</li> </ul>	<ul style="list-style-type: none"> <li>• Modelos entidad relación y simulación de sistemas.</li> <li>• Sistemas computacionales: dispositivos, <i>hardware</i> y <i>software</i>, y resolución de problemas.</li> </ul>	<ul style="list-style-type: none"> <li>• Redes e Internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.</li> <li>• Algoritmos y programación: Lenguajes, métodos y entornos de programación.</li> </ul>	<ul style="list-style-type: none"> <li>• Algoritmos y programación: Lenguajes, métodos y entornos de programación.</li> <li>• Representación y análisis de datos.</li> <li>• Impacto social de la computación: interacciones, cultura, seguridad y ética.</li> </ul>
Tiempo Estimado	6 semanas (12 horas)	6 semanas (12 horas)	6 semanas (12 horas)	6 semanas (12 horas)

## Módulo Obligatorio 1

### Visión panorámica

#### Gran idea

Los problemas pueden descomponerse en partes más pequeñas y manejables, organizarse siguiendo patrones y solucionarse en base a reglas, modelos y algoritmos.

#### Objetivos de aprendizaje

**OA1.** Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que lo componen. (Descomponer)

**OA2.** Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. (Evaluar)

**OA3.** Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. (Modelar)

#### Conocimientos esenciales

- Procesos y funcionamiento de las cosas para resolver problemas cotidianos.
- Modelos entidad-relación y simulación de sistemas.

#### Tiempo estimado

6 semanas (12 horas)

## Propósito Módulo obligatorio 1

En el módulo 1 de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media se espera que los estudiantes comprendan que, frente a cualquier situación es necesario entender el problema y hacer un análisis previo de lo que considera el mismo, para poder dar la mejor respuesta posible y satisfacer así las necesidades de forma eficiente y óptima. Asimismo, todo problema puede resolverse por medio de una serie de algoritmos, por ejemplo: los seres humanos en sus rutinas cotidianas al momento de tener que resolver problemas, siguen una serie de acciones, conscientes o inconscientes, en las que evalúan, organizan o clasifican para llegar a una solución, estas acciones pueden constituir perfectamente un algoritmo. Dicho esto, en este módulo los estudiantes distinguirán que la pieza básica del algoritmo es la descomposición y que un problema puede descomponerse en partes pequeñas y abordables que lo conforman en su generalidad.

Los Objetivos de Aprendizaje del módulo 1 buscan desarrollar las habilidades de descomposición y evaluación a través del modelamiento de base de datos, y para comprender estos conceptos desde las ciencias de la computación, los estudiantes pueden organizar la información de manera efectiva y eficiente, pues, al hacer un modelo de base de datos, ellos también descomponen información para darle un sentido. Podemos entender en este módulo que el algoritmo y la base de datos no son tan diferentes, pues ambas pueden ser descompuestas, evaluadas y modeladas.

Los Objetivos de Aprendizaje del módulo 1 desarrollan las actitudes del siglo XXI: Maneras de pensar, Maneras de vivir el mundo y las Herramientas para trabajar, promoviendo la utilización de las herramientas y uso de la información para resolver problemas, diseñar soluciones e interactuar con sistemas en diferentes contextos de la vida diaria. Finalmente, las actividades propuestas son una oportunidad que promueve la ludificación del espacio de aprendizaje desde contextos reales y cercanos para el uso de las tecnologías y la alfabetización digital.

## Ruta de Aprendizaje del Módulo obligatorio 1

### ¿Cómo descomponer problemas para evaluarlos usando diferentes fuentes de información, modelos y herramientas?

**Actividad de desempeño 1:** Construyen algoritmos en base a la descomposición de los problemas que surgen en la cotidianidad, al tiempo que evalúan las respuestas y toman decisiones respecto de si son las más acertadas.

**Actividad de desempeño 2:** Evalúan las mejores opciones al momento de resolver un problema teniendo en cuenta el algoritmo y pseudocódigo. Además, aprovechan oportunidades para poder utilizar códigos para resolver situaciones-problema.



**Actividad de desempeño 3:** Modelan bases de datos en contextos que plantean la necesidad de soluciones con apertura a diferentes perspectivas y clasifican y descomponen objetos cotidianos haciendo una relación simple entre ellos.

#### Actividad de desempeño

**4:** Resuelven problemas en base a la entidad-relación entre tablas de acuerdo con el modelamiento de bases de datos, fundamentando cada atributo en base al dato que almacenará.

## Actividad de desempeño 1

### Propósito de la actividad

Esta actividad se dedicará a los fundamentos básicos del funcionamiento del algoritmo. La meta consiste en poder descomponer problemas en base a la lógica computacional de un algoritmo y llevar a cabo esas instrucciones a una codificación centralizada para resolver el problema principal.

### Objetivos de Aprendizaje

**OA1.** Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que lo componen. **(Descomponer)**

**OA2.** Organizar datos de una situación problemática, estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

### Conocimiento esencial

- Procesos y funcionamiento de las cosas para resolver problemas cotidianos.

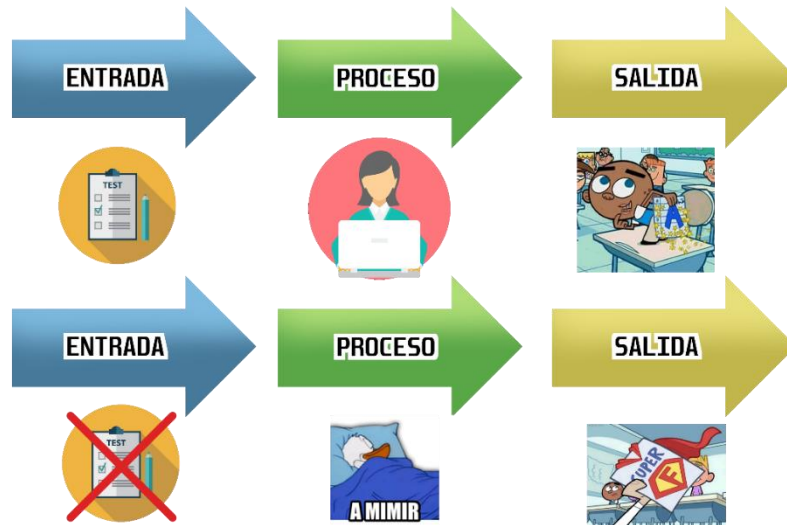
### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

El docente presenta a los estudiantes dos casos: el de Carolina, quien estudió mucho para un examen muy importante de Matemática, y el caso de José, quien, en vez de estudiar para dicho examen, decidió dormir. Ambos casos se presentan de forma algorítmica donde se muestra el resultado de sus decisiones:

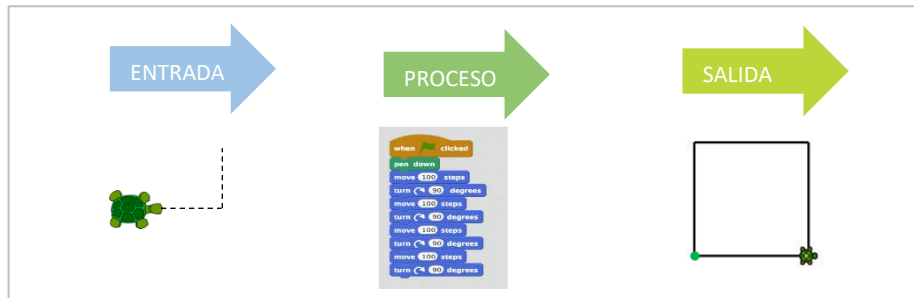


Algunas de las preguntas que pueden promover el proceso de comunicar una argumentación sobre el tipo de conductas de los dos sujetos:

- ¿Qué algoritmo representa la aprobación del examen?
- ¿Cuál es el proceso que hizo Carolina?
- ¿Cuál es el proceso que hizo José?
- ¿Cómo se podría hacer un algoritmo donde uno elige si dormir o no?
- ¿Cuál sería la salida de Carolina y José, respectivamente?

### Construcción de conocimiento

Para construir el conocimiento sobre los pasos del algoritmo (entrada, proceso y salida) y su descomposición, se sugiere comenzar con un acercamiento del proceso de entrada, proceso y salida, utilizando el lenguaje verbal o bien un programa en bloques, describiendo qué se tiene por entrada, cuál sería el proceso y qué se obtiene en la salida.




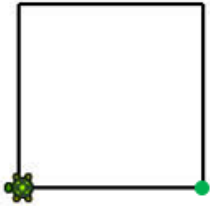
<b>ENTRADA</b>	La tortuga Kira parte en el punto verde.
<b>PROCESO</b>	<ul style="list-style-type: none"> <li>• La tortuga avanza en LINEA RECTA 100 pasos HACIA LA DERECHA</li> <li>• La tortuga se gira en 90°</li> <li>• La tortuga avanza en LINEA RECTA 100 pasos</li> <li>• La tortuga se gira en 90°</li> <li>• La tortuga avanza en LINEA RECTA 100 pasos</li> <li>• La tortuga se gira en 90°</li> <li>• La tortuga avanza en LINEA RECTA 100 pasos</li> <li>• La tortuga se gira en 90°</li> <li>• La tortuga se para.</li> </ul> <p>Los ángulos se entienden en el sentido positivo (contra el reloj).</p>
<b>SALIDA</b>	La tortuga Kira SE ENCUENTRA EN el punto verde.

Para motivar la creación y elaboración de un seudocódigo, se sugiere considerar los movimientos y algunas convenciones del lenguaje, por ejemplo, utilizar solo tres mayúsculas para los movimientos: AVA para avanzar, RET para retroceder, GIR para girar; incluir paréntesis para indicar con números enteros la cantidad de pasos y para indicar una rotación se utiliza una coma para indicar la unidad de medida en grados.

<b>ENTRADA</b>	La tortuga Kira parte en el punto verde.
<b>PROCESO</b>	<ul style="list-style-type: none"> <li>• AVA (100)</li> <li>• GIR (90, grados)</li> <li>• AVA (100)</li> <li>• GIR (90, grados)</li> <li>• AVA (100)</li> <li>• GIR (90, grados)</li> <li>• AVA (100)</li> <li>• GIR (90, grados)</li> </ul> <p>Los ángulos se entienden en el sentido positivo (contra el reloj).</p>
<b>SALIDA</b>	La tortuga Kira SE ENCUENTRA EN el punto verde.

Se sugiere comparar este seudocódigo con la programación escrita en inglés en bloques en SCRATCH y ver diferencias o similitudes entre ambos lenguajes.

¿Qué similitudes o diferencias hay entre el pseudocódigo y el programa?

Seudocódigo	Programa en bloques
<ul style="list-style-type: none"> <li>• AVA (100)</li> <li>• GIR (90, grados)</li> <li>• AVA (100)</li> <li>• GIR (90, grados)</li> <li>• AVA (100)</li> <li>• GIR (90, grados)</li> <li>• AVA (100)</li> <li>• GIR (90, grados)</li> </ul> <p>Los ángulos se entienden en el sentido positivo (contra el reloj).</p>	 <p>The image shows a Scratch script starting with a 'when clicked' event block, followed by a 'pen down' block, and then four pairs of 'move 100 steps' and 'turn 90 degrees' blocks. The turn blocks have a counter-clockwise arrow icon.</p>
<p>- Coincide en los movimientos de avanzar en 4 ocasiones y girar en 90 grados en 4 ocasiones.</p> <p>- Difieren en el sentido del giro que va en el sentido del reloj y la tortuga va en otra dirección. En la imagen se ve que difieren también del punto de partida.</p> <div style="text-align: center;">  </div> <p>En la instrucción del programa, se indica el punto de partida como la instrucción “<i>pen down</i>” que significa “poner la punta del lápiz en el papel”.</p>	



### Práctica guiada

Se sugiere elaborar un pseudocódigo y las nociones de entrada, proceso y salida para algunos procedimientos matemáticos que son utilizados cotidianamente, por ejemplo, el cálculo del promedio de notas.

¿Cuál sería mi promedio final en este semestre?

Teniendo en cuenta que:

- El promedio de las 3 calificaciones parciales es de un 60% y que mis notas son 4,8 – 6,0 – 5,7. Notas que se pueden escribir como 48 – 60 – 57 para facilitar la lectura.
- El examen final es de un 40% del total y tengo un 6,8 que también se puede escribir como 68
- Las fórmulas para considerar son:

$$\text{Promedio Nota} = (\text{Sumar notas}) / \text{Cantidad de notas}$$

$$\text{Promedio final} = (\text{Promedio Nota} * 0.6) + (\text{Examen Final} * 0.4)$$

Para guiar a los estudiantes en la elaboración de respuestas, se sugiere que utilicen los términos de entrada, proceso y salida para resolver la problemática planteada, elaborar un pseudocódigo con lo planteado anteriormente y así poder analizar las diferencias entre escribir los pasos de un algoritmo y un pseudocódigo, el cual podría hacerse mediante PSeint<sup>19</sup>, que es un *software* de escritura de códigos, o bien utilizar una hoja de papel.

Acción	Resultado esperado			Preguntas de apoyo
<b>Confeccionar una lista para las fases de un algoritmo</b>	ENTRADA	PROCESO	SALIDA	¿Qué fórmula hay que utilizar? ¿Qué datos se deben ingresar en la planilla de procesos? ¿Cuál formato será más conveniente para los datos?
	Promedio de calificaciones parciales 60% (48 – 60 – 57)  Examen final 40% (68)	1. Promedio de notas = $(N1+N2+N3) / 3$  2. Promedio de notas $\cdot 0.6 +$ NF $\cdot 0.4$  3. Tener el resultado	Resultado numérico: promedio final	

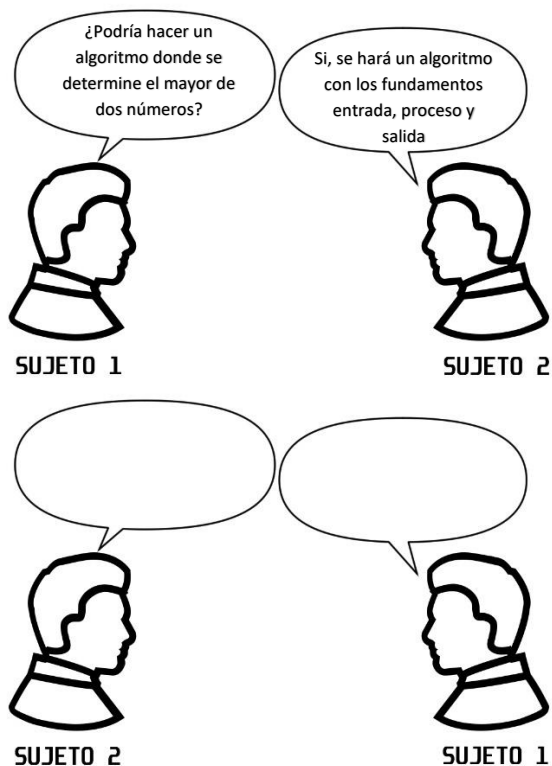
<sup>19</sup> *Software* para primeros pasos de programación usados en el programa: <https://link.curriculumnacional.cl/http://pseint.sourceforge.net/>

Acción	Resultado esperado	Preguntas de apoyo
<p>Elaborar un pseudocódigo relacionado con lo expresado en las fases de los algoritmos hechos anteriormente</p>	<pre> <b>Algoritmo</b> CalcularPromedioFinal  <b>Definir</b> N1, N2, N3, NF <b>como entero</b>; <b>Definir</b> PromedioNota, PromedioFinal <b>como real</b>;  N1 = 48; N2 = 60; N3 = 57; NF = 68;  <b>Escribir</b> "Notas: " N1 " - " N2 " - " N3;  PromedioNota = (N1+N2+N3)/3;  <b>Escribir</b> "Promedio de notas: " PromedioNota;  <b>Escribir</b> "Nota final: " NF;  PromedioFinal = PromedioNota * 0.6 + NF * 0.4  <b>Escribir</b> "Promedio de notas final: " PromedioFinal;  <b>FinAlgoritmo</b> </pre>	<p>¿Cómo empezar con la codificación?</p> <p>¿Cómo organizar todo para que sea coherente?</p> <p>¿Cómo podemos aplicar las fórmulas en el pseudocódigo?</p>
<p>Evaluar la eficiencia del código y ver si es posible que uno ingrese las notas</p>	<pre> <b>Algoritmo</b> CalcularPromedioFinal  <b>Definir</b> N1, N2, N3, NF <b>como entero</b>; <b>Definir</b> PromedioNota, PromedioFinal <b>como real</b>;  <b>Escribir</b> "Ingrese nota 1"; <b>leer</b> N1;  <b>Escribir</b> "Ingrese nota 2"; <b>leer</b> N2;  <b>Escribir</b> "Ingrese nota 3"; <b>leer</b> N3;  PromedioNota = (N1+N2+N3)/3;  <b>Escribir</b> "Promedio de notas: " PromedioNota;  <b>Escribir</b> "Ingrese nota final"; <b>leer</b> NF;  PromedioFinal = PromedioNota * 0.6 + NF * 0.4  <b>Escribir</b> "Promedio de notas final: " PromedioFinal;  <b>FinAlgoritmo</b> </pre>	<p>¿Con esto entendemos las partes en qué se descomponen los algoritmos?</p> <p>¿Cómo explicamos que el pseudocódigo presentado es el correcto?</p> <p>¿Cómo podemos verificar si el código funciona?</p> <p>¿Para qué nos sirve este algoritmo o</p>

Acción	Resultado esperado	Preguntas de apoyo
		programación en seudocódigo si podemos usar a cada vez una calculadora?

### Práctica independiente

Se sugiere hacer un juego de roles en parejas, donde uno será el cliente que solicita un programa o solución algorítmica al otro. La solución se construye entre ambos y se pueden elaborar instrucciones escritas completas, o bien elaborar un pseudocódigo o trabajar con Pseudint; en todos los casos se sugiere identificar la entrada, el proceso y la salida.



¿Podría hacer un algoritmo donde la tortuga camine en espiral?

Si, en pseudocódigo, probaremos dos entradas diferentes, pero el mismo proceso y salida.

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Entrada-proceso y salida	Diseña una fase de algoritmo simple.	Diseña fases de algoritmo que son suficientes.	Diseña fases de algoritmo organizadas y entendibles, apreciando cada detalle solicitado.
Elaboración de pseudocódigos	Elabora un pseudocódigo.	Elabora un pseudocódigo que corresponde medianamente a lo solicitado.	Elabora un pseudocódigo que corresponde a lo solicitado.

## Orientaciones al docente

**Para unificar conceptos disciplinares:** Entendemos que evaluar algoritmos en esta actividad requiere de algunos conocimientos previos y entender algunos conceptos principales para lograr que los estudiantes comprendan lo que es un algoritmo y cuáles son sus aplicaciones en PSeint. Es recomendable dar explicaciones simples y el concepto básico de las fases de un algoritmo; a su vez, se explica con mayor detalle el uso de las condicionales “si” y “según”.

El concepto base en esta actividad corresponde a una serie de instrucciones que siguen patrones o reglamentos que se realizan inconscientemente. A partir de esto será más fácil poder ampliar las demás funcionalidades, siendo una herramienta para expresar algoritmos más complejos que tengan toma de decisiones.

- **Seudocódigo:** Es un código falso, es decir, es una idea de cómo sería en base al lenguaje de programación (una base), para ello se utiliza PSeint o Scratch.
- **Código:** Es la codificación de un algoritmo en base a un lenguaje de programación profesional como C#, JAVA o Python.

Las nociones de pseudocódigo funcional y óptimo pueden ser trabajadas a medida que los estudiantes ejecutan el programa de otro grupo y comparten sus pseudocódigos con otros para que los ejecuten manualmente.

**Actitudes:** Para apoyar el desarrollo de las actividades es recomendable utilizar en ciertos periodos computadores (PSeint) para que los estudiantes puedan ver si sus creaciones son correctas y verifiquen si puede haber más alternativas a sus problemas, pero siempre está la alternativa de hacer los pseudocódigos mediante una hoja de papel o en un documento Word. También se pueden incluir algunas preguntas que fortalezcan la toma de decisiones del mismo algoritmo y así contribuyan al bien común, por ejemplo: ¿Qué proceso sería el más eficaz?, ¿será la opción más acertada? o ¿es posible hacerlo de otra manera? En cuanto a desarrollar los trabajos colaborativos, se sugiere promover un trabajo simultáneo donde los estudiantes se ayuden mutuamente a conseguir el objetivo de ambos.

En la situación experiencial, Carolina estudió y logró sacarse una buena calificación, pero en cuanto a José, él prefirió quedarse dormido en vez de estudiar y por eso tuvo una mala calificación en el examen de matemáticas.

**Orientaciones para organizar e implementar el juego de roles:** se sugieren las siguientes indicaciones para guiar la implementación del trabajo de pares con un juego de roles:

- Elabore tarjetas con dos roles: el estudiante que pregunta puede ser considerado el cliente y la persona que responde el informático. Indicar en las tarjetas lo que se espera como tarea en cada rol.
- Ambos tendrán sus tarjetas y empezarán al mismo tiempo. Cada uno dirá el problema y después elaborarán un algoritmo basado en la problemática.
- Determine el tiempo necesario para elaborar un algoritmo simple para comenzar.
- Organice la clase y el espacio para el trabajo en pares.

- Determine el tiempo para que ambos estudiantes revisen el algoritmo del otro. Deben dar una retroalimentación de acuerdo con el seguimiento de la tarjeta, como si falta algún detalle pequeño o si se puede hacer de mejor manera.
- Luego, determinar el tiempo para que ambos se ayuden mutuamente para hacer un pseudocódigo del algoritmo que acaban de hacer.
- Determine lo que se hará con los resultados y con la entrega de las respuestas. No es necesario que se haga una presentación de los resultados. Se sugiere conversar sobre las dificultades y obstáculos que surgen a la hora de elaborar un algoritmo y/o pseudocódigo.

## Anexo

Elaborar una calculadora en Pseint con las funciones de sumar, restar, multiplicar y dividir con solo dos valores a elegir para realizar el cálculo.

Acción	Resultado esperado		
	ENTRADA	PROCESO	SALIDA
<p><b>Confeccionar una lista para las fases de un algoritmo</b></p>	<p><b>Calculadora</b></p>	<ol style="list-style-type: none"> <li>1. Seleccionar qué opción usará (sumar, restar, multiplicar o dividir)</li> <li>2. Ingresar los dos valores que se quieren calcular. "Valor 1 (opción elegida) Valor 2"</li> <li>3. Calcular la operación matemática. "Valor 1 (opción elegida) Valor 2 = Total"</li> </ol>	<p>Resultado de la operación</p>

**Algoritmo** Calculadora

```

//definir variables
Definir opc, valor1, valor2 Como Entero;
Definir total como real;
//entrada
Escribir "Calculadora";
Escribir "1: Sumar";
Escribir "2: Restar";
Escribir "3: Multiplicar";
Escribir "4: Dividir";
Escribir "Ingrese opcion:";
Leer opc;

Si opc ≥ 1 y opc ≤ 4 Entonces
    Escribir "Ingrese valor 1: ";
    Leer valor1;

    Escribir "Ingrese valor 2: ";
    Leer valor2;
    Si opc = 1 Entonces
        //proceso
        total = valor1 + valor2;
        //salida
        Escribir "La suma de " valor1 " + " valor2 " = " total;
    SiNo
        Si opc = 2 Entonces
            //proceso
            total = valor1 - valor2;
            //salida
            Escribir "La resta de " valor1 " - " valor2 " = " total;
        SiNo
            Si opc = 3 Entonces
                //proceso
                total = valor1 * valor2;
                //salida
                Escribir "La multiplicacion de " valor1 " * " valor2 " = " total;
            SiNo
                Si opc = 4 Entonces
                    //proceso
                    total = valor1 / valor2;
                    //salida
                    Escribir "La division de " valor1 " / " valor2 " = " total;
                FinSi
            FinSi
        FinSi
    FinSi
    //salida
    Escribir "Opcion invalida";
FinSi

```

Elaborar un código relacionado con lo expresado en las fases de los algoritmos hechos anteriormente

**FinAlgoritmo**

**Algoritmo** Calculadora

```

//definir variables
Definir opc, valor1, valor2 Como Entero;
Definir total como real;
//entrada
Escribir "Calculadora";
Escribir "1: Sumar";
Escribir "2: Restar";
Escribir "3: Multiplicar";
Escribir "4: Dividir";
Escribir "Ingrese opcion:";
Leer opc;

Si opc ≥ 1 y opc ≤ 4 Entonces
    Escribir "Ingrese valor 1: ";
    Leer valor1;

    Escribir "Ingrese valor 2: ";
    Leer valor2;

    Segun opc
        1:
            //proceso
            total = valor1 + valor2;
            //salida
            Escribir "La suma de " valor1 " + " valor2 " = " total;
        2:
            //proceso
            total = valor1 - valor2;
            //salida
            Escribir "La resta de " valor1 " - " valor2 " = " total;
        3:
            //proceso
            total = valor1 * valor2;
            //salida
            Escribir "La multiplicacion de " valor1 " * " valor2 " = " total;
        4:
            //proceso
            total = valor1 / valor2;
            //salida
            Escribir "La division de " valor1 " / " valor2 " = " total;
    FinSegun
SiNo
    //salida
    Escribir "Opcion invalida";
FinSi

FinAlgoritmo

```

Evaluar la eficiencia y optimización del código



## Actividad de desempeño 2

### Propósito

En esta actividad se pone énfasis en la evaluación para que los estudiantes tengan una base sólida de los conceptos principales del algoritmo y así desenvolverse de manera más fluida con los términos ya aprendidos, además de añadir algunos conceptos nuevos que entreguen oportunidades cuando se quiera expresar algoritmos en un código para resolver un problema.

### Objetivos de Aprendizaje

**OA2.** Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función.  
**(Evaluar)**

### Conocimiento esencial

- Procesos y funcionamiento de las cosas para resolver problemas cotidianos.

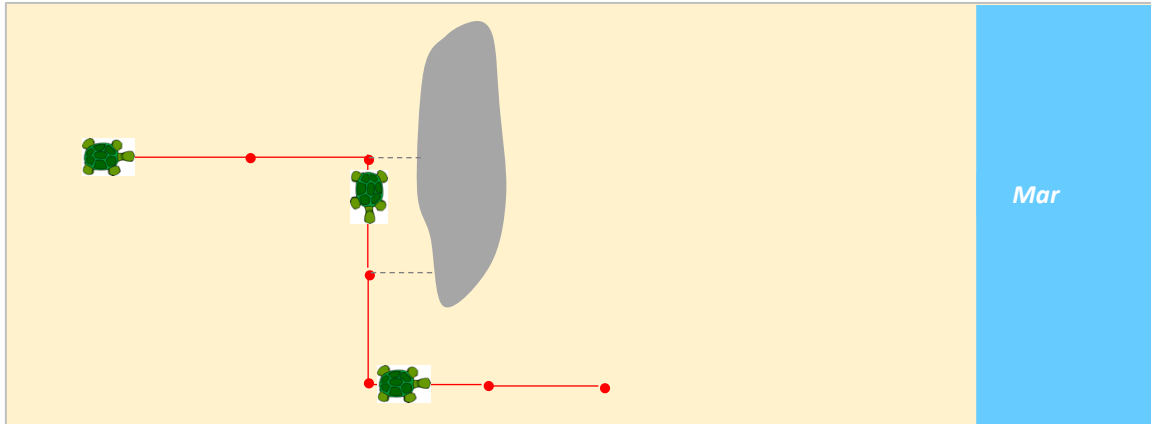
### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

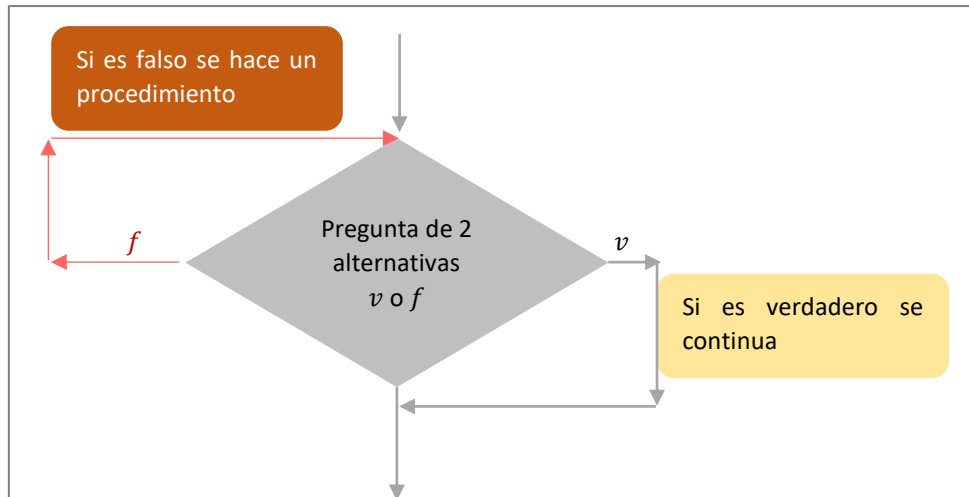
El docente presenta una situación que puede ser transferida a situaciones reales y a un uso cotidiano de “si pasa esto” y/o de “según esto o lo otro”, introduciendo las nociones con una situación de decisión entre 2 alternativas para el camino de la tortuga.



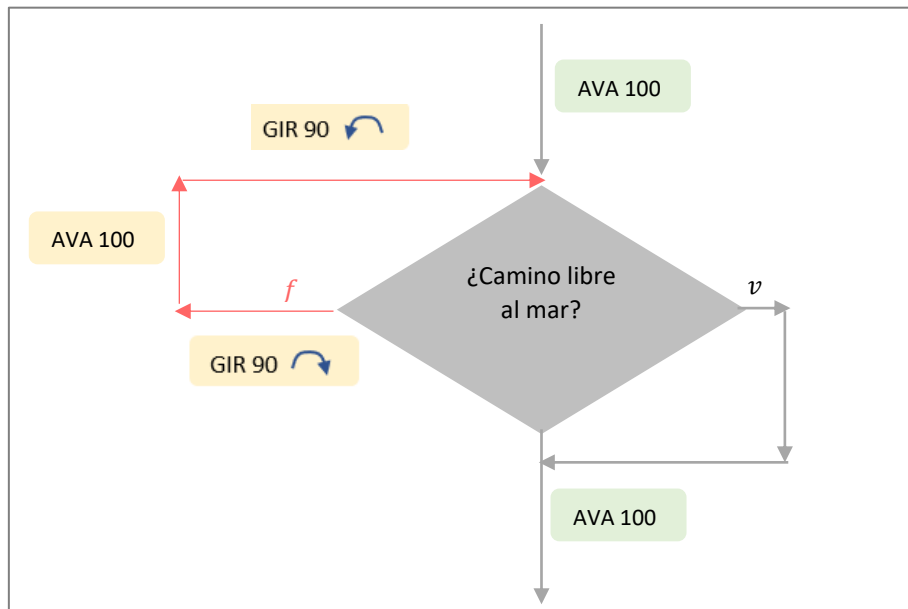
La tortuga Kira quiere llegar al mar. En su camino avanza en pasos de 100, pero puede ocurrir que un obstáculo como una roca esté en su camino. La tortuga puede desviar este obstáculo girando en 90° a la derecha o a la izquierda y luego avanzando 100 pasos, girando 90° a la izquierda y mirando otra vez si el camino está despejado para el próximo avance de 100 pasos hacia el mar y repitiendo el procedimiento en el caso de que “no” sea así.

### Construcción de conocimiento

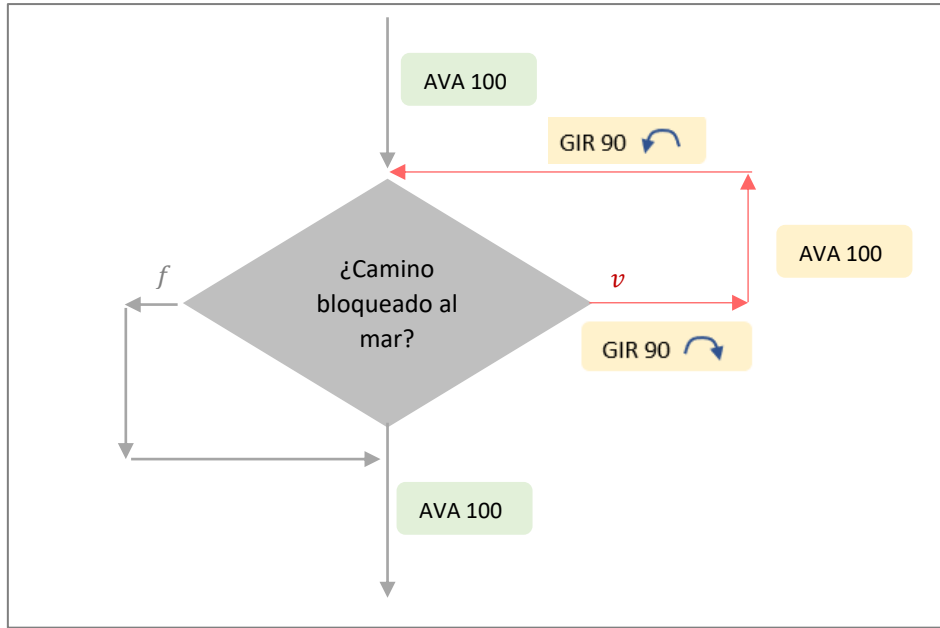
Para construir el conocimiento, se sugiere resaltar que en programación todos los lenguajes formales disponen de una orden que requiere una decisión entre 2 alternativas, las que pueden ser “sí o no”, o bien “verdadero o falso”. Esta instrucción, en la que se decide entre 2 alternativas, se presenta en el recuadro de abajo con la forma de un rombo.



Se sugiere ejemplificar esta idea de 2 alternativas utilizando el camino de la tortuga, esto es, esquematisando el programa y utilizando el pseudocódigo creado en la actividad de desempeño anterior, precisando la instrucción de decisión que se refiere a la situación de la tortuga con un rombo.



Se sugiere verificar de forma activa si este programa lleva a la tortuga al mar. Para esto, los estudiantes pueden actuar el esquema o bien utilizar fichas o material concreto. También, se sugiere probar con un segundo esquema de programación, en el cual se cambia la pregunta decisiva en el rombo por “¿Camino bloqueado al mar?”.



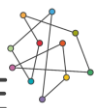
### Práctica guiada

Se sugiere guiar la verificación de que los algoritmos propuestos son en cierta medida la mejor opción, o bien presentar la forma de cambiarlos, si es necesario y así cumplir con las necesidades que se solicitan. El estudiante también podría proponer mejoras para dicho algoritmo y así conseguir satisfacer y dar respuestas a las necesidades.

Con respecto a lo anterior se trabajará con un menú donde habrá dos opciones a elegir:

- Verificar si un número es par o impar.
- Una tabla de multiplicar.

Acción	Resultado esperado	Preguntas de apoyo
<p>Elaborar un código lo más fiel posible a lo establecido en el problema</p>	<pre> <b>Algoritmo</b> ParImpar   <b>Definir</b> aux, num, NumMayor, variables <b>Como Entero</b>    <b>Escribir</b> 'Seleccione que tipo de cliente es: ';   <b>Escribir</b> '1) Ver si un numero es par o impar';   <b>Escribir</b> '2) Tabla de multipliucar';   <b>Escribir</b> '3) Ver cual de los 10 numeros es el mayor';   <b>Leer</b> aux;    <b>Segun</b> aux <b>Hacer</b>   1:     <b>Escribir</b> "Escriba un número: "     <b>Leer</b> num      <b>Si</b> num % 2 = 0 <b>Entonces</b>       <b>Escribir</b> num, " es un numero par"     <b>SiNo</b>       <b>Escribir</b> num, " es un numero impar"     <b>Fin Si</b>    2:     <b>Escribir</b> "Ingrese un Número del 1 - 10 : "     <b>Leer</b> Num      <b>Escribir</b> 1 " X " Num " = " (1 * Num)     <b>Escribir</b> 2 " X " Num " = " (2 * Num)     <b>Escribir</b> 3 " X " Num " = " (3 * Num)     <b>Escribir</b> 4 " X " Num " = " (4 * Num)     <b>Escribir</b> 5 " X " Num " = " (5 * Num)     <b>Escribir</b> 6 " X " Num " = " (6 * Num)     <b>Escribir</b> 7 " X " Num " = " (7 * Num)     <b>Escribir</b> 8 " X " Num " = " (8 * Num)     <b>Escribir</b> 9 " X " Num " = " (9 * Num)     <b>Escribir</b> 10 " X " Num " = " (10 * Num)    <b>FinSegun</b> <b>FinAlgoritmo</b>           </pre>	<p>¿Realmente este código cumple con lo prometido de la mejor manera?</p> <p>¿Existe alguna otra forma de hacerlo?</p> <p>¿Cómo podemos saber si un número es par o impar en un código?</p>



<p>Después del análisis y la nueva metodología aprendida, ¿cómo arreglaría o mejoraría el anterior código?</p>	<pre> <b>Algoritmo</b> ParImpar   <b>Definir</b> aux, num, NumMayor, variables <b>Como Entero</b>    <b>Escribir</b> 'Seleccione que tipo de cliente es: ';   <b>Escribir</b> '1) Ver si un numero es par o impar';   <b>Escribir</b> '2) Tabla de multipliucar';   <b>Escribir</b> '3) Ver cual de los 10 numeros es el mayor';   <b>Leer</b> aux;    <b>Segun</b> aux <b>Hacer</b>   1:     <b>Escribir</b> "Escriba un número: "     <b>Leer</b> num      <b>Si</b> num % 2 = 0 <b>Entonces</b>     ..... <b>Escribir</b> num, " es un numero par"     <b>SiNo</b>     ..... <b>Escribir</b> num, " es un numero impar"     <b>Fin Si</b>   2:     <b>Escribir</b> "Ingrese un Número del 1 - 12 : "     <b>Leer</b> Num      <b>Para</b> Cont ←1 <b>Hasta</b> 12 <b>Con Paso</b> 1 <b>Hacer</b>     ..... <b>Escribir</b> Cont " X " Num " = " (Cont * Num)     <b>Fin Para</b>   <b>FinSegun</b> <b>FinAlgoritmo</b> </pre>	<p>¿Cómo sé que esta es la mejor opción?</p> <p>¿Cuál sería la mayor diferencia entre el código antiguo y el nuevo?</p> <p>¿De qué otra manera se podría usar la nueva metodología?</p>
--	--	---

### Práctica independiente

Teniendo en cuenta el anterior ejercicio y viendo que el resultado es satisfactorio, se decide implementar una nueva opción llamada verificar cuál de los 10 números es el mayor. Esta opción servirá para aprender a usar estas nuevas herramientas y así poder expandir la resolución del problema anterior:

Acción	Resultado esperado
<p>Elaborar un código donde se añada la nueva opción ya mencionada en el código hecho anteriormente.</p>	<pre> <b>Algoritmo</b> ParImpar   <b>Definir</b> aux, Num, NumMayor, variables <b>Como Entero</b>    <b>Escribir</b> 'Seleccione que tipo de cliente es: ';   <b>Escribir</b> '1) Ver si un numero es par o impar';   <b>Escribir</b> '2) Tabla de multipliucar';   <b>Escribir</b> '3) Ver cual de los 10 numeros es el mayor';   <b>Leer</b> aux;    <b>Segun</b> aux <b>Hacer</b>   1:     <b>Escribir</b> "Escriba un número: "     <b>Leer</b> Num      <b>Si</b> Num % 2 = 0 <b>Entonces</b>         <b>Escribir</b> Num, " es un numero par"     <b>SiNo</b>         <b>Escribir</b> Num, " es un numero impar"     <b>Fin Si</b>    2:     <b>Escribir</b> "Ingrese un Número del 1 - 12 : "     <b>Leer</b> Num      <b>Para</b> Cont ←1 <b>Hasta</b> 12 <b>Con Paso</b> 1 <b>Hacer</b>         <b>Escribir</b> Cont " X " Num " = " (Cont * Num)     <b>Fin Para</b>    3:     <b>Para</b> Cont ← 1 <b>Hasta</b> 10 <b>Con Paso</b> 1 <b>Hacer</b>         <b>Escribir</b> "Escriba un número: "         <b>Leer</b> Num          <b>Si</b> NumMayor &lt; Num <b>Entonces</b>             NumMayor = Num         <b>Fin Si</b>     <b>Fin Para</b>      <b>Escribir</b> "El Mayor es : ", NumMayor    <b>FinSegun</b> <b>FinAlgoritmo</b> </pre>

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

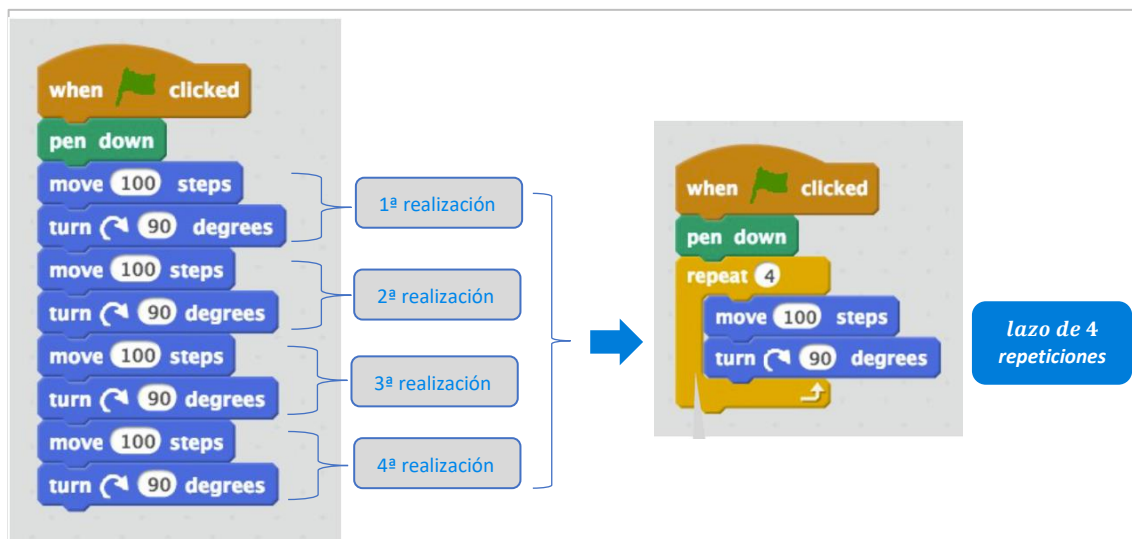
Criterio	Inicial	Intermedio	Avanzado
Códigos	Elabora un código.	Elabora un código que cumple con lo establecido.	Elabora un código que cumple con lo establecido y también lo analizó y mejoró en el proceso.
Metodologías	Usa las metodologías aprendidas.	Usa las nuevas estrategias de forma medianamente efectiva.	Usa las nuevas estrategias de forma efectiva.

### Orientaciones al docente

**Para unificar conceptos disciplinares:** Podemos entender el funcionamiento de un algoritmo y también cómo se tienen que resolver los problemas para tener la respuesta más aceptable y óptima posible. Se sugiere integrar nuevas estrategias, como “parar” y “repetir”.

- Parar: permite medir cuantas veces queremos que se repita la misma acción.
- Repetir: permite repetir hasta que una acción en concreto se cumpla.

Estas son herramientas para complementar nuestro entendimiento en el funcionamiento del algoritmo y la codificación. El objetivo principal es hacer más sencillo el trabajo de elaboración de programas y la ejecución de ellos. Se sugiere indicar en qué momentos existen pasos iguales y si se puede utilizar la iteración del procedimiento.

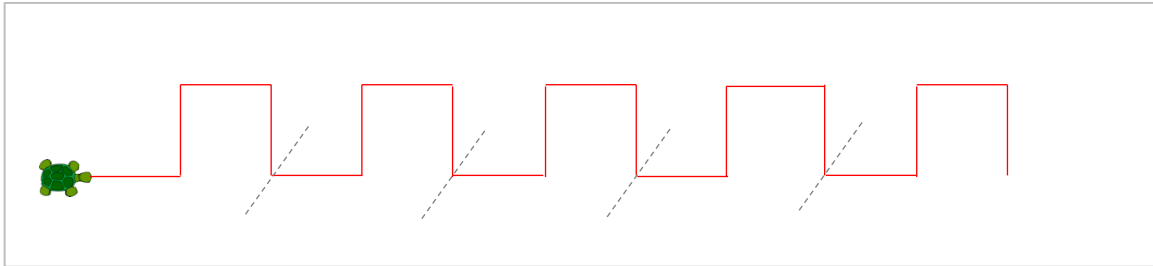




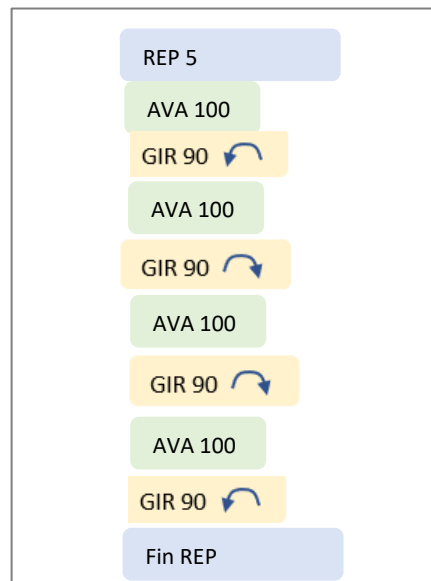
Algunas preguntas guías para esta introducción de los ciclos o bucles podrían ser:

- ¿Cuáles son los pasos de desplazamiento y giro que se repiten?
- ¿Cómo se está abreviando la escritura del programa?

Se sugiere también ver estos ciclos de forma visual, indicando qué es lo que se repite y la cantidad de veces que se repite. Por ejemplo, en el caso del camino de la tortuga.



Relacionando con la programación en seudocódigo:



**Actitudes:** Siempre que se esté usando un computador a la hora de codificar un algoritmo, se recomienda verificar el código de una manera más eficiente y, también, poder ejecutar y ver su funcionamiento, aprovechando siempre las herramientas disponibles. Para el desarrollo de los trabajos colaborativos, es recomendable que trabajen en equipo para que se ayuden mutuamente; también es útil para que surjan conversaciones y se tomen decisiones para evaluar de forma correcta el algoritmo usando criterios de función y funcionamiento.

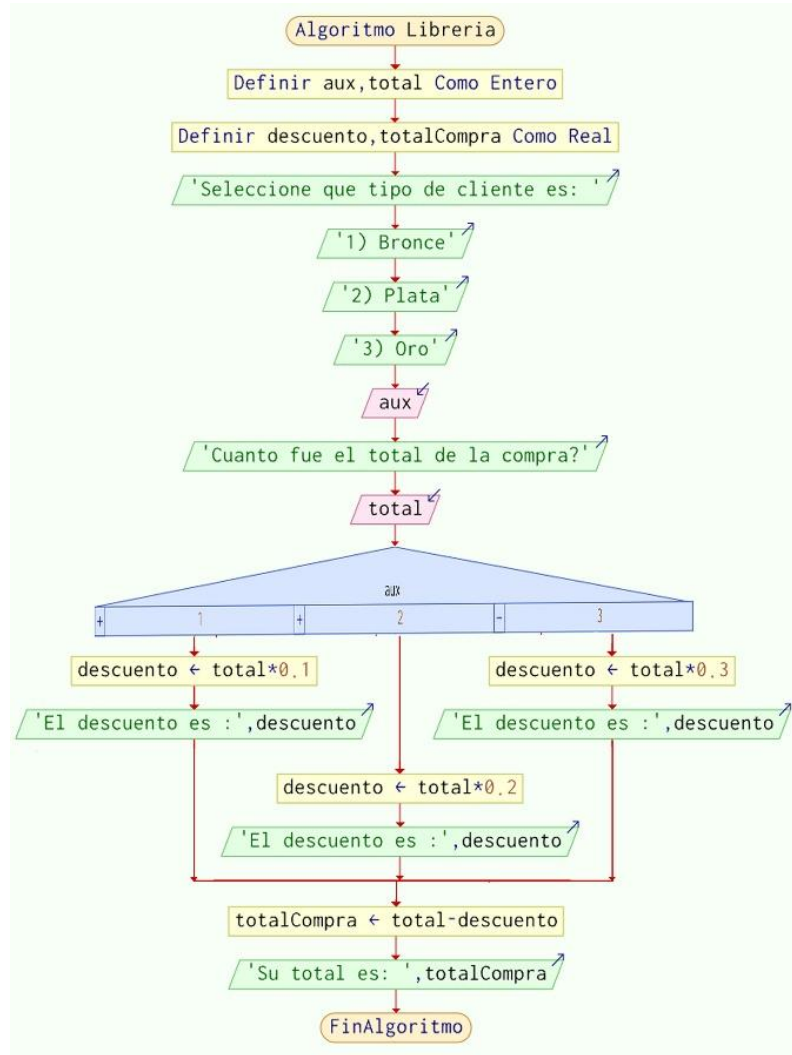
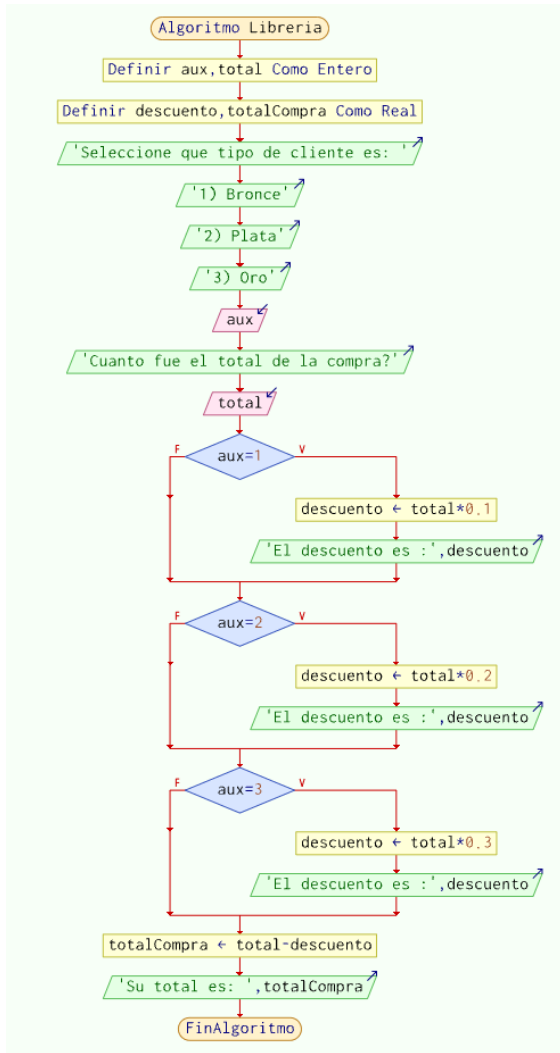
**Orientaciones para organizar e implementar el trabajo en equipo:** Se sugiere las siguientes indicaciones para guiar la implementación del trabajo en parejas o tríos:

- Elabore ejercicios donde se tengan en cuenta las funcionalidades “para” y “repetir”.
- Cada pareja o trío tendrá su ejercicio a desarrollar.
- Determine el tiempo necesario de desarrollo del código.
- Los alumnos analizan y verifican si todo está correcto y ven si se le puede hacer una mejora a lo propuesto en el problema.
- Determine el tiempo para el análisis y corrección de código.

## Anexo

En esta actividad se muestran dos esquemas de programas donde los estudiantes pueden encontrar similitudes y diferencias. Se sugiere guiar la actividad por medio de las siguientes preguntas:

- ¿Cuál es el problema para cada algoritmo?
- ¿Habrá una forma más eficiente de elaborar estos algoritmos?
- ¿Cuáles serían las diferencias entre los dos algoritmos en cuanto a funcionalidad?
- ¿Cómo se pueden mejorar estos algoritmos?



La siguiente actividad complementa la práctica independiente y marca nuevamente los procedimientos que se repiten en el esquema de programación:

<p>El código es eficiente, pero tener que escribir los 10 números en cada momento es agotador. Sería mejor si la máquina hiciera ese trabajo y nos mostrara los números que escogió.</p>	<pre> 3: Para Cont ← 1 Hasta 10 Con Paso 1 Hacer     Num ← Aleatorio(0,100)     Escribir "El numero " Cont " es: " Num     Si NumMayor &lt; Num Entonces         NumMayor = Num     Fin Si Fin Para  Escribir "El Mayor es : ", NumMayor FinSegun         </pre>
--	--

Mucho mejor, pero a la hora de finalizar el problema, se termina de ejecutar el código. ¿Cómo se podría hacer un bucle para poder elegir todas las opciones y uno decidir cuándo termina de correr el código?

```

1  Algoritmo ParImpar
2  Definir aux, Num, NumMayor, variables Como Entero
3
4  Repetir
5      Escribir 'Seleccione que tipo de cliente es: ';
6      Escribir '1) Ver si un numero es par o impar';
7      Escribir '2) Tabla de multipliucar';
8      Escribir '3) Ver cual de los 10 numeros es el mayor';
9      Escribir '4) Salir';
10     Leer aux;
11
12     Segun aux Hacer
13         1:
14             Escribir "Escriba un número: "
15             Leer Num
16
17             Si Num % 2 = 0 Entonces
18                 Escribir Num, " es un numero par"
19             SiNo
20                 Escribir Num, " es un numero impar"
21             Fin Si
22         2:
23             Escribir "Ingrese un Número del 1 - 12 : "
24             Leer Num
25
26             Para Cont ← 1 Hasta 12 Con Paso 1 Hacer
27                 Escribir Cont " X " Num " = " (Cont * Num)
28             Fin Para
29         3:
30             Para Cont ← 1 Hasta 10 Con Paso 1 Hacer
31                 Num ← Aleatorio(0,100)
32                 Escribir "El numero " Cont " es: " Num
33                 Si NumMayor < Num Entonces
34                     NumMayor = Num
35                 Fin Si
36             Fin Para
37
38             Escribir "El Mayor es : ", NumMayor
39     FinSegun
40     Hasta Que aux = 4
41 FinAlgoritmo

```

## Actividad de desempeño 3

### Propósito

Esta actividad, está enfocada en el modelamiento de base de datos y los inicios del sistema de clasificación de estructuración de esta. Se identificarán las entidades y atributos descomponiendo puntos clave de los problemas a la vez que aplicarán, a través de actividades concretas, las diferentes clasificaciones de atributos que constituye un modelo de base de datos.

### Objetivos de Aprendizaje

**OA1.** Formular problemas a partir de un contexto o situación específica, con apertura a distintas perspectivas, identificando las partes que lo componen. **(Descomponer)**

**OA3.** Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

### Conocimiento esencial

- Modelos entidad-relación y simulación de sistemas.

### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

El docente presenta a los estudiantes dos imágenes para identificar ciertos parámetros que se deben saber y necesitar a la hora de modelar una base de datos.



Algunas de las preguntas que pueden promover el proceso de comunicar una argumentación sobre lo que podemos observar son:

- ¿Cuáles serían las entidades de las siguientes imágenes?
- ¿Qué atributos tendría cada entidad?
- ¿Cómo se relacionarían las dos entidades?

### Construcción de conocimiento

Para construir el conocimiento sobre cómo armar un modelo de base de datos, se sugiere empezar por analizar objetos simples para entender lo es una entidad y un atributo, con sus respectivas descripciones. Primero, se le mostrará a los estudiantes imágenes para descomponer y así clasificarlas en torno al funcionamiento de una base de datos.



Para guiar a los estudiantes en la elaboración de respuestas, se sugiere que utilicen los términos “atributo” y “entidad” para la resolución del problema. También, se sugiere ampliar e identificar los atributos, decidiendo si son obligatorios, opcionales o claves primarias, para esto, se pueden considerar como mínimo 3 atributos.

Acción	Resultado esperado		Preguntas de apoyo
<p>Confeccionar una lista para la clasificación de las entidades</p>	ENTIDAD	ATRIBUTO	<p>¿Cómo identificar cada atributo?</p> <p>¿Son todos los atributos que se pudieron encontrar?</p> <p>¿Puede haber datos que se repitan en cada entidad?</p> <p>¿Existe alguna forma de relacionar todas las entidades?</p>
	AVIÓN	Alas Alerón Estabilizador Cabina Motor Asiento Piloto Pasajero	
	BARCO	Timón Asiento Ancla Cubierta Capitán Tripulación Mapa	
	ÁRBOL	Tronco Hojas Fruta Nido de pájaro	
	PLANETA	Tierra Agua Personas Animales Capa de ozono Rocas Árboles Casas Países	



Acción	Resultado esperado	Preguntas de apoyo
<p>Distinguir las diferentes clasificaciones de atributos y verificar si en los atributos existe alguno que sirva como identificador</p>	<div style="display: flex; flex-wrap: wrap;"> <div style="border: 1px solid blue; padding: 5px; margin: 5px; width: 50%;"> <p style="text-align: center;"><b>AVION</b></p> <ul style="list-style-type: none"> <li># *Codigo</li> <li>* alas</li> <li>* aleron</li> <li>* estabilizador</li> <li>* cabina</li> <li>* motor</li> <li>o asiento</li> <li>* piloto</li> <li>o pasajero</li> </ul> </div> <div style="border: 1px solid blue; padding: 5px; margin: 5px; width: 50%;"> <p style="text-align: center;"><b>BARCO</b></p> <ul style="list-style-type: none"> <li># *codigo</li> <li>* timon</li> <li>o asiento</li> <li>* ancla</li> <li>* cubierta</li> <li>* capitán</li> <li>o tripulacion</li> <li>o mapa</li> </ul> </div> <div style="border: 1px solid blue; padding: 5px; margin: 5px; width: 50%;"> <p style="text-align: center;"><b>ARBOL</b></p> <ul style="list-style-type: none"> <li># * nombre</li> <li>* tronco</li> <li>o ramas</li> <li>o hojas</li> <li>o fruta</li> <li>o nido_pajaros</li> </ul> </div> <div style="border: 1px solid blue; padding: 5px; margin: 5px; width: 50%;"> <p style="text-align: center;"><b>PLANETA</b></p> <ul style="list-style-type: none"> <li># * nombre</li> <li>o tierra</li> <li>o agua</li> <li>o personas</li> <li>o animales</li> <li>o capa_ozono</li> <li>o rocas</li> <li>o arbol</li> <li>o casa</li> <li>o pais</li> </ul> </div> </div>	<p>¿Qué hacer si no le encuentro una clave primaria?</p> <p>¿Cómo identificar una clave primaria?</p> <p>¿Cómo saber si un atributo es obligatorio u opcional?</p>
<p>Relacione las tablas entre ellas</p>	<pre> graph TD     PLANETA[PLANETA] --&gt; ARBOL[ARBOL]     PLANETA --&gt; AVION[AVION]     PLANETA --&gt; BARCO[BARCO] </pre>	<p>¿Cómo saber si están relacionadas?</p>

### Práctica guiada

Para guiar la evaluación del modelamiento de base de datos se debe tener claro cómo definir cada tabla que se vaya generando durante la lectura del mismo problema. Es importante tener presente que no siempre será fácil identificar todas las entidades, atributos y relaciones, pero es necesario tener una base firme para después consultar las dudas que uno pueda tener. Así, el objetivo de esta práctica será crear un modelo de base de datos relacionado con la aplicación Facebook o similar.

**Regístrate** ✕

Es rápido y fácil.

Nombre Apellido

Número de celular o correo electrónico

Contraseña nueva

Fecha de nacimiento ?

12 ago 2021

Sexo ?

Mujer  Hombre  Personalizado

Al hacer clic en "Regístrate", aceptas nuestras [Condiciones](#), la [Política de datos](#) y la [Política de cookies](#). Es posible que te enviemos notificaciones por SMS, que puedes desactivar cuando quieras.

**Regístrate**

Correo electrónico o número de teléfono

Contraseña

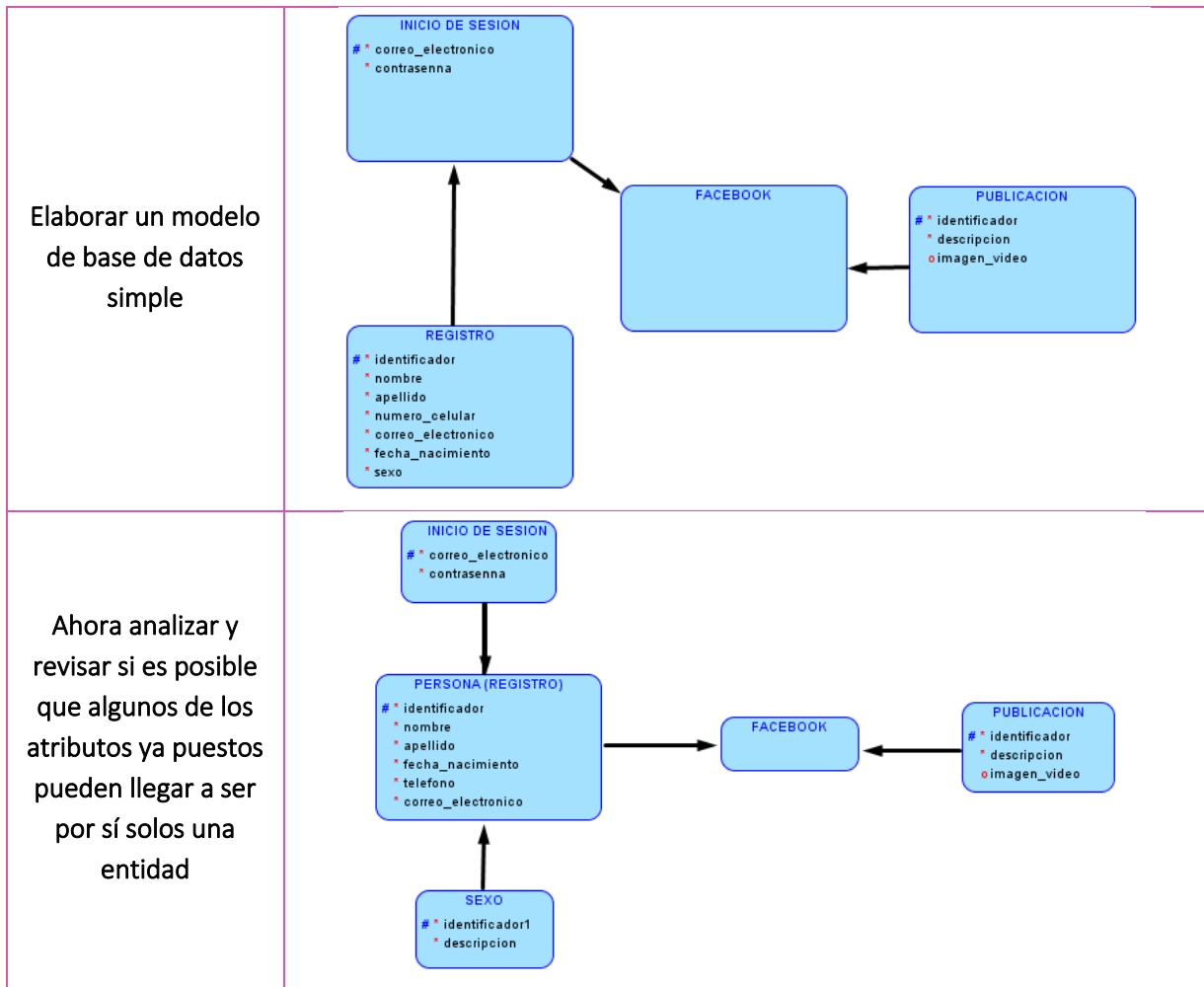
**Iniciar sesión**

[¿Olvidaste tu contraseña?](#)

**Crear cuenta nueva**

Al completar, el docente aclarará que no se está solicitando un modelo extremadamente complejo en ese sentido; solo se solicitan elementos básicos que están relacionados en este registro de cuenta, por ejemplo, nombre, apellido, correo, contraseña y su respectivo inicio de sesión, o bien la publicación y el registro.

Acción	Resultado esperado	
<p>Identificar las características que tiene Facebook y revisar si son claves primarias, obligatorias u opcionales</p>	<p><b>FACEBOOK</b></p>	<p><b>INICIO DE SESION</b></p> <ul style="list-style-type: none"> <li># * correo_electronico</li> <li>* contrasenna</li> </ul>
	<p><b>PUBLICACION</b></p> <ul style="list-style-type: none"> <li># * identificador</li> <li>* descripcion</li> <li>o imagen_video</li> </ul>	<p><b>REGISTRO</b></p> <ul style="list-style-type: none"> <li># * identificador</li> <li>* nombre</li> <li>* apellido</li> <li>* numero_celular</li> <li>* correo_electronico</li> <li>* fecha_nacimiento</li> <li>* sexo</li> </ul>



### Práctica independiente

Se sugiere una actividad individual donde el estudiante tiene que analizar un caso con el objetivo de identificar cuáles son las entidades y atributos de una situación de venta de productos. Se sugiere mostrar una imagen expositiva de la situación y que detalle los pormenores del problema. El docente **guía** la actividad para que el estudiante tenga suficiente tiempo para su respectivo desarrollo y **orienta** el análisis, retroalimentando cada punto de las respuestas.

Un pequeño ejemplo sería que marcará con **AZUL** las entidades, con **ROJO** los atributos obligatorios y con **VERDE** los atributos opcionales.

Los productos que se venden en el almacén son comprados a distintos **proveedores** de Valparaíso y Viña del Mar. A algunos proveedores se les llama por **teléfono** y se conoce su **Rut** y **nombre**. Un proveedor puede vender distintos **productos**; es necesario almacenar **el precio de venta** que ofrece cada proveedor y la **fecha de validez** del precio ofrecido.

De los **productos**, se registra el **identificador del código de barra, descripción y precio unitario de venta sin incluir el IVA y clasificación del producto**; de **algunos productos se sabe la fecha de vencimiento y capacidad del envase**; de **todos los productos se conoce el nombre de la marca y el identificador de esta**. Un **producto** se relaciona sólo con una **marca**.

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Clasificación de tablas	Identifica las tablas.	Identifica la mayoría de las tablas, aunque faltan atributos.	Clasifica todas las tablas con sus respectivos atributos y relaciones.
Diferenciación de atributos	Diferencia solo algunos atributos.	Diferencia medianamente los atributos.	Diferencia todos los atributos (obligatorios y opcionales).
Relacionar tablas	Relaciona algunas tablas.	Relaciona las tablas con algunos errores.	Relaciona todas las tablas sin errores.

## Orientaciones al docente

**Para unificar conceptos disciplinares:** Entendemos que el modelamiento de base de datos es algo más complejo que simplemente hacer un mapa conceptual, pero tampoco está tan alejado de esa realidad. La realización de un modelo de base de datos es para poder tener un mejor control de la información que se guardará dentro de esta base de datos. Se sugiere explicar los elementos básicos primordiales de un modelo de base de datos.

- El modelo contará con una entidad por cada uno de los componentes del proceso de negocio.
- Los atributos, un componente fundamental de cada modelo entidad-relación, nos permiten describir las propiedades que tiene cada entidad.
  - **Obligatorios:** la instancia del atributo debe tener un valor válido.
  - **Opcionales:** la instancia del atributo puede o no tener un valor.
- Con las relaciones se establecen vínculos entre parejas de entidades.
- La clave o llave primaria es un atributo o combinación de atributos que identifican de forma exclusiva una y solo una instancia de una entidad; ningún subconjunto del modelo entidad-relación la posee.

Se sugiere organizar el diagrama entidad-relación (DER) desde sus componentes básicos hasta los más complejos, comenzando de manera intuitiva con una entidad y sus atributos para luego avanzar con dos entidades. Según el contexto elegido, se pueden incluir las nociones de atributos compuestos o simples, como también el grado de una relación.

**Actitudes:** Para apoyar el desarrollo de las actividades no es necesario en esta etapa inicial el uso de computadores. Se recomienda que, a la hora de modelar una base de datos, se utilice una planilla de cálculo y que, al momento de utilizar lenguaje de programación, también se utilicen los computadores. Por otra parte, es recomendable retroalimentar a los estudiantes considerando una apertura a diferentes perspectivas y considerando con claridad si lo que uno hace está bien o mal ejecutado. En cuanto a la actividad individual, es preferible que se favorezca la autonomía, para identificar cada característica de la situación y hacer que el estudiante tenga independencia y confianza en el entendimiento de la entidad-relación.

**Orientaciones para organizar e implementar el trabajo individual:** Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Elaborar un ejercicio amplio que esté centrado en algún sistema de negocio o minimercado, para que los alumnos identifiquen las tablas necesarias. Se sugiere pensar también en las relaciones que se dan la situación para luego utilizarlas en las próximas actividades.
- Determine un tiempo para poder analizar y revisar todo el problema; para identificar las tablas que esconde.
- Pedir a los estudiantes revisar todas las respuestas que escogieron. Esto beneficiará a los que no lograron identificar todo, además de poder aclarar dudas de cómo poder identificar cada caso.

## Actividad de desempeño 4

### Propósito

En esta actividad se reforzará y ampliará el conocimiento del modelamiento de base de datos, en el que el estudiante aprenderá a cómo relacionar de forma correcta las tablas entre sí y, a su vez, la normalización de estas. También, los estudiantes aprenderán la estrategia de tres formas comunes para mantener un orden al momento de crear tablas.

### Objetivos de Aprendizaje

**OA2.** Organizar datos de una situación problemática, estableciendo secuencias e identificando fallos fallas y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

**OA3.** Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

### Conocimiento esencial

Modelos entidad-relación y simulación de sistemas.

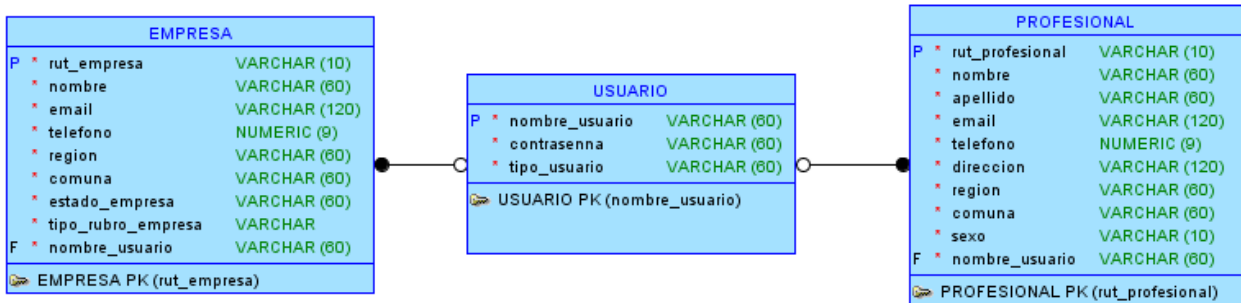
### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

El docente presenta a los estudiantes un caso particular de un modelo de base de datos que se ve simple a primera vista, pero tiene sus fallas a la hora de ser algo funcional y no es algo efectivo y eficiente que se esperaría de una base de datos funcional.



Algunas preguntas para motivar a los estudiantes a indicar y fundamentar los errores apreciables en este modelo de base de datos:

- ¿Qué atributos podrían formar su propia tabla? (entidad y atributos)
- ¿Qué significan los caracteres verdes al lado de los atributos?
- ¿Qué significa la “F” en el último atributo de la tabla “Empresa” y “Profesional”?

### Construcción de conocimiento

Para construir el conocimiento sobre la forma correcta de hacer un modelo de base de datos, se sugiere leer los problemas cuidadosamente e intuir la información que quizás no esté complementada. El estudiante lee con atención y descifra las entidades, atributos y relaciones que debe tener el texto.

Marcará de color AZUL las entidades, de color ROJO los atributos obligatorios, de color NARANJA los atributos opcionales y de color VERDE los verbos que le permitan reconocer relaciones entre entidades.

#### Caso Habitación

*Un adulto con un sueldo promedio tiene un cuarto bastante grande que convertirá en la habitación de su hijo. Las pertenencias que tendrá dicha habitación serán una o más de una cama con sábanas, almohadas y puede que lleve un cubrecama y un peluche. También, tendrá una repisa donde quizás se guardarán libros, videojuegos y discos de música, pero la repisa también tendrá algunos juguetes. Una habitación puede tener más de una repisa.*

*La habitación tendrá un escritorio donde se pondrá una televisión de 32 pulgadas, un notebook, un mouse, y si alcanza, puede que también alguna videoconsola. La habitación puede tener más de un escritorio. También tendrá un armario donde se guardará ropa interior, poleras y pantalones; puede que también se guarden calcetines, polerones y camisas. Puede tener un armario o más.*

*Dentro de la habitación habrá un tomacorriente, también tendrá un color, un piso de alfombra y puede que tenga un futón. Por último, objetos tales como libros, videojuegos y música tendrán su nombre,*

*clasificación de edad (videojuegos) y género. Los libros, videojuegos y música pueden estar en más de una repisa, al igual que una repisa puede tener muchos libros, videojuegos y música.*

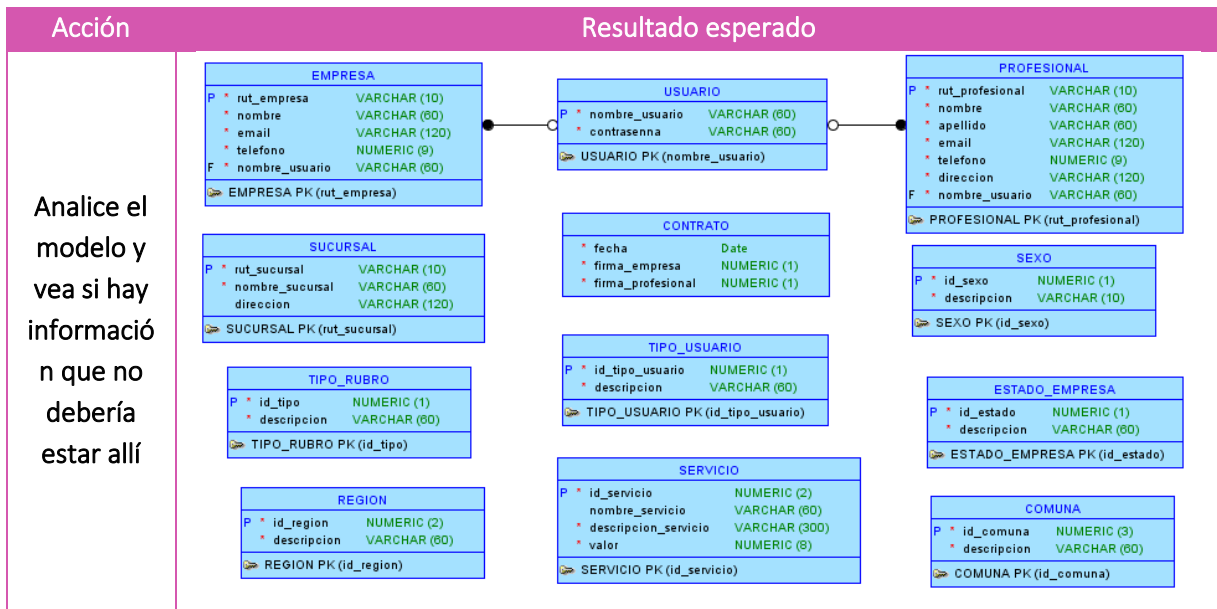
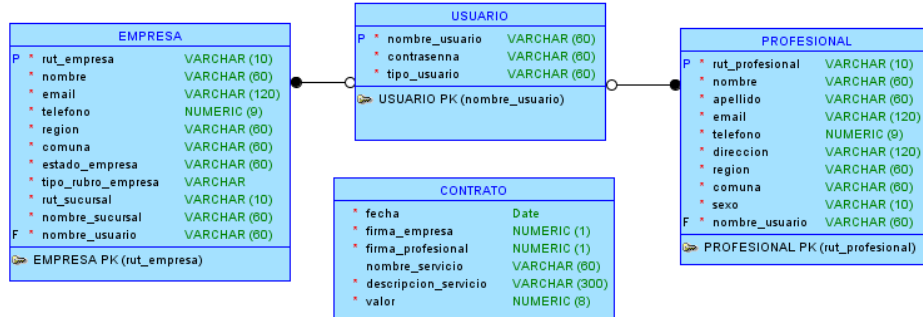
Acción	Resultado esperado	Preguntas de apoyo
<p>Marcará de color <b>AZUL</b> las entidades, de color <b>ROJO</b> los atributos obligatorios, de color <b>NARANJA</b> los atributos opcionales y de color <b>VERDE</b> los verbos que le permitan reconocer relaciones entre entidades.</p>	<p>Un adulto con un sueldo promedio tiene un cuarto bastante grande que convertirá en la habitación de su hijo. Las pertenencias que tendrá dicha habitación serán una o más de una <b>cama</b> con <b>sábanas, almohadas</b> y <b>puede que lleve un cubrecama y un peluche</b>. También tendrá una <b>repisa</b> donde quizás se guardarán libros, videojuegos y discos de música, pero la repisa <b>si tendrá algunos juguetes</b>. Una habitación puede tener más de una repisa.</p> <p>La habitación tendrá un <b>escritorio</b> donde se pondrá una <b>televisión de 32 pulgadas, un notebook, un mouse</b> y si alcanza <b>puede que también alguna videoconsola</b>. La habitación puede tener más de un escritorio. También tendrá un <b>armario</b> donde se guardará <b>ropa interior, poleras y pantalones</b>; <b>puede que también se guarden calcetines, polerones y camisas</b>. Puede tener un armario o más.</p> <p>Dentro de la <b>habitación</b> habrá un <b>tomacorriente, también tendrá un color, un piso de alfombra</b> y <b>puede que tenga un futón</b>. Por último, objetos tales como <b>libros, videojuegos y música</b> tendrán su <b>nombre, clasificación de edad (videojuegos) y género</b>. Los libros, videojuegos y música pueden estar en más de una repisa, al igual que una repisa puede tener muchos libros, videojuegos y música.</p>	<p>¿Cómo determinar qué dato equivale a que dato?</p> <p>¿Cómo identificar las relaciones?</p> <p>¿Qué sucede si hay relaciones que no se mencionan?</p>

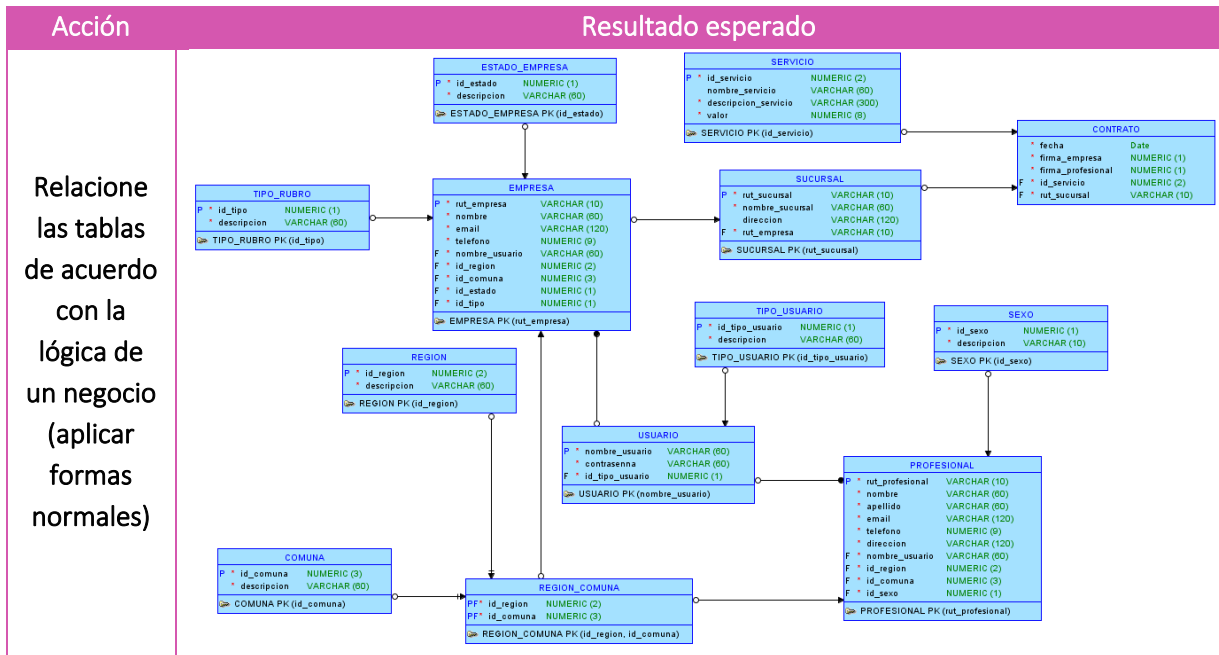


Acción	Resultado esperado	Preguntas de apoyo
<p>Modele la base de datos que acaba de clasificar en el problema</p>		<p>¿Cómo saber si las relaciones se hicieron bien?</p>
<p>Analicé y evalué si se le pueden hacer mejoras al modelo de base de datos (aplicar formas normales)</p>		<p>¿Cómo saber qué atributo podría llevar su propia tabla? ¿Cómo saber cuándo aplicar las formas normales?</p>

## Práctica guiada

Para poder guiar la evaluación del modelamiento de base de datos, es necesario analizar los problemas y dar el mejor resultado, al igual que con los algoritmos. Para ello, el docente realizará el ejemplo que utilizamos anteriormente, pero ahora con una tabla adicional.





### Práctica independiente

Se sugiere una actividad individual en la que tengan que analizar un caso y ver cuáles son las entidades, atributos y relaciones de esta. Se puede utilizar el ejercicio de la anterior actividad de práctica independiente o uno nuevo. De cualquier modo, lo ideal es siempre analizar punto por punto junto a la clase y que después hagan el respectivo modelo de base de datos individualmente.

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Analizar problemas	Analiza el problema.	Analiza el problema y consigue obtener la mayoría de los datos.	Analiza el problema y logra identificar todos los datos de la información.
Relacionar tablas	Relaciona las tablas con errores.	Relaciona las tablas con algunos errores.	Logra relacionar las tablas sin errores.
Formas normales	Aplica las formas normales con errores.	Aplica las formas normales con algunos desaciertos.	Aplica las formas normales de buena manera, además de saber cuándo ya es suficiente.

## Orientaciones al docente

**Para unificar conceptos disciplinares:** Entenderemos que la normalización de la base de datos es una técnica de modelado que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas del paso del diagrama entidad-relación al modelo relacional. Se entiende que las relaciones y normalizar un modelo es primordial en esta actividad. Se sugiere explicar y dar ejemplos para poder entender de mejor manera las tres formas principales de relaciones (1:1 – 1:M – M:N). Además, es necesario entender el concepto de las tres formas normales y en qué circunstancias se utilizan. Se puede utilizar el siguiente ejemplo:

- Primera forma normal: los atributos contienen valores simples.

En la siguiente tabla hay un atributo con doble valor, no es simple. Entonces, esto se puede solucionar eliminando el atributo o se incluye un nuevo atributo como segundo correo.

ID	Nombre	Puesto	Salario	Correo electrónico
12	Fabiola Reyes	Secretaria	200	fr@empresa.com
13	Elsa Fredes	Secretaria	200	<a href="mailto:elf@empresa.com">elf@empresa.com</a> ; <a href="mailto:f@empresa.gma">f@empresa.gma</a> x
14	Ani Lindo	Administrativa	150	al@empresa.com

- Segunda forma normal: los atributos contienen valores simples y todos los atributos que no son clave primaria tienen dependencia funcional completa con respecto a todas las claves existentes en el esquema. Para recuperar un atributo, se accede por la clave completa.

ID	Nombre	Puesto	Salario	Correo electrónico
12	Fabiola Reyes	Secretaria	200	fr@empresa.com
13	Elsa Fredes	Secretaria	200	<a href="mailto:elf@empresa.com">elf@empresa.com</a>
13	Elsa Fredes	Secretaria	200	<a href="mailto:f@empresa.gma">f@empresa.gma</a>
14	Ani Lindo	Administrativa	150	al@empresa.com

Para hacer una segunda forma normal, se tienen dos tablas posibles ID-nombre-puesto-salario:

ID	Nombre	Puesto	Salario
12	Fabiola Reyes	Secretaria	200
13	Elsa Fredes	Secretaria	200
14	Ani Lindo	Administrativa	150

La otra es ID-correo:

ID	Correo electrónico
12	fr@empresa.com
13	<a href="mailto:elf@empresa.com">elf@empresa.com</a>
13	<a href="mailto:f@empresa.gma">f@empresa.gma</a>
14	al@empresa.com

- Tercera forma normal: está en primera y segunda forma normal. Además, los atributos que no son clave no dependen transitivamente.

ID	Nombre	Puesto	Salario
12	Fabiola Reyes	Secretaria	200
13	Elsa Fredes	Secretaria	200
14	Ani Lindo	Administrativa	150

La tabla tiene dependencias transitivas ID-puesto y puesto-salario, entonces ID-salario. Para normalizar esta tabla hay dos posibilidades para separar: la primera tabla es puesto y salario:

Puesto	Salario
Secretaria	200
Administrativa	150

Y la segunda tabla es ID-nombre-puesto, considerando el puesto como una clave foránea.

ID	Nombre	Puesto
12	Fabiola Reyes	Secretaria
13	Elsa Fredes	Secretaria
14	Ani Lindo	Administrativa

**Actitudes:** Para apoyar el desarrollo de las actividades, es recomendable la utilización de computadores a la hora de modelar la base de datos, para que así obtengan una retroalimentación más visual de lo correcto e incorrecto de lo que están desarrollando. Además, se debe ser constante a la hora de hacer un modelo de base de datos, ya que las bases aprendidas en la anterior actividad deben estar solucionadas y entendidas por parte del estudiante. Se sugiere que el docente motive al estudiante para que pueda realizar preguntas con honestidad que lo ayuden a comprender los elementos faltantes para lograr una resolución. En cuanto al desarrollo de los trabajos individuales, se sugiere promover la autonomía y pensamiento propio del estudiante y también la confianza en sus conocimientos previos.

**Orientaciones para organizar e implementar trabajo individual:** Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Elaborar un ejercicio amplio que esté centrado en algún sistema de negocio o minimercado, para que los alumnos identifiquen las tablas necesarias. Se sugiere pensar también en las relaciones que se dan la situación para luego utilizarlas en las próximas actividades.
- Determinar un tiempo para poder analizar y revisar todo el problema, para identificar las tablas que esconde.
- Llamar a los estudiantes a revisar entre todas las respuestas que escogieron. Esto beneficiará a los que no lograron identificarlas todas, además de poder aclarar dudas de cómo identificar cada caso. Una vez finalizada la retroalimentación con los estudiantes, deberán trabajar en las mejoras del modelamiento de la situación e incluir aquellas relaciones o pasos que son necesarios. Puede ser en una hoja, pero se recomienda siempre el uso de computadores para estos casos.
- Determinar el tiempo para desarrollar el modelo de base de datos con sus respectivas entidades, atributos y relaciones, y aplicar las formas normales.
- Dar retroalimentación del ejercicio llamando a algún estudiante o simplemente mostrarles al final de la clase cómo debería ser el modelo y por qué debería quedar así; explicar cómo lo analizó y desarrollo.

## Módulo obligatorio 2

### Visión panorámica

<p><b>Gran idea</b></p> <p>El comportamiento de un sistema computacional (dispositivos, <i>hardware</i> y <i>software</i>) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea con el fin de ejecutar una acción o resolver un problema.</p>
<p style="text-align: center;"><b>Objetivos de Aprendizaje</b></p> <p><b>OA2.</b> Organizar datos de una situación problemática, estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. <b>(Evaluar)</b></p> <p><b>OA3.</b> Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. <b>(Modelar)</b></p>
<p style="text-align: center;"><b>Conocimientos esenciales</b></p> <ul style="list-style-type: none"> <li>• Modelos entidad-relación y simulación de sistemas.</li> <li>• Sistemas computacionales: dispositivos, <i>hardware</i> y <i>software</i>, y resolución de problemas.</li> </ul>
<p style="text-align: center;"><b>Tiempo estimado</b></p> <p style="text-align: center;">6 semanas (12 horas)</p>

## Propósito Módulo obligatorio 2

En el módulo 2 de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media, se espera que los estudiantes refuercen aprendizajes previos y que de forma inicial comiencen a comprender que organizar y modelar datos está asociado a las instrucciones que ellos le den a un sistema computacional. Esto con el objetivo de recibir una respuesta deseada, incorporando, a su vez, que a través del uso de dispositivos, *hardware* y *software* diferentes se automatizan desempeños de acuerdo con las tareas que deseen realizar. La idea de la relación entre el ser humano y la máquina se refuerza desde los niveles de interacción que tengan los estudiantes con herramientas digitales en cada actividad de desempeño para diseñar una solución y resolver un problema.

Los estudiantes desarrollan nuevos conocimientos y profundizan en habilidades para resolver problemas cada vez más complejos que requieren una organización y modelado de datos más precisos y óptimos. en el proceso la identificación de fallos, se debe considerar el uso criterios de usabilidad y simulaciones simples, además de evaluar las tecnologías digitales que tengan disponibles para entregar soluciones acordes las necesidades que se plantean. El énfasis del módulo está en que los estudiantes confíen en sus capacidades para interactuar con uno o más sistemas computacionales y la evaluación es una habilidad que necesitan seguir trabajando para asegurar que una solución, ya sea un algoritmo, un sistema o un proceso, cumpla eficientemente con su propósito de conseguir una respuesta de acuerdo con lo requerido.

Los Objetivos de Aprendizaje del módulo 2 desarrollan las actitudes del siglo XXI del ámbito de las Maneras de pensar y Herramientas para trabajar, promoviendo el uso provechoso de las herramientas para resolver problemas o diseñar soluciones, pensando con apertura en distintas perspectivas y contextos. Finalmente, las actividades de desempeño propuestas esperan ser una oportunidad para comprender el uso de las tecnologías y cómo las personas interactúan con ellas de forma cotidiana para satisfacer necesidades.



## Ruta de aprendizaje del Módulo obligatorio 2

### ¿Cómo se modelan los niveles de interacción entre los sistemas computacionales y los sistemas sociales, ambientales o industriales que enfrenta un usuario?

#### Actividad de Desempeño 1:

Construyen modelos de base de datos utilizando entidades, atributos y relaciones para obtener diferentes formas de resolver problemas.

#### Actividad de

**Desempeño 2:** Resuelven problemas utilizando los condicionales, mientras, repetir y para, aprovechando las herramientas para analizar y evaluar un problema.



#### Actividad de Desempeño 3:

Modelan y evalúan subprocesos, implementando de acuerdo con las necesidades del problema las funcionalidades ya conocidas.

#### Actividad de Desempeño

**4:** Modelan y evalúan formas para programar utilizando 1, 2 y 3 dimensiones, y resolviendo problemas de algoritmos.

## Actividad de desempeño 1

### Propósito

Esta actividad está orientada a consolidar los conocimientos relacionados con el modelo de base de datos y, a su vez, conocer la nueva entidad de relación extendida con la finalidad de tener un mayor número de herramientas y formas de modelar y realizar una base de datos de forma correcta. Estas nuevas formas son elementos extras que pueden cambiar de forma significativa el resultado final de un proceso para resolver un problema.

### Objetivos de Aprendizaje

**OA3.** Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles.

**(Modelar)**

### Conocimiento esencial

Modelos entidad-relación y simulación de sistemas.

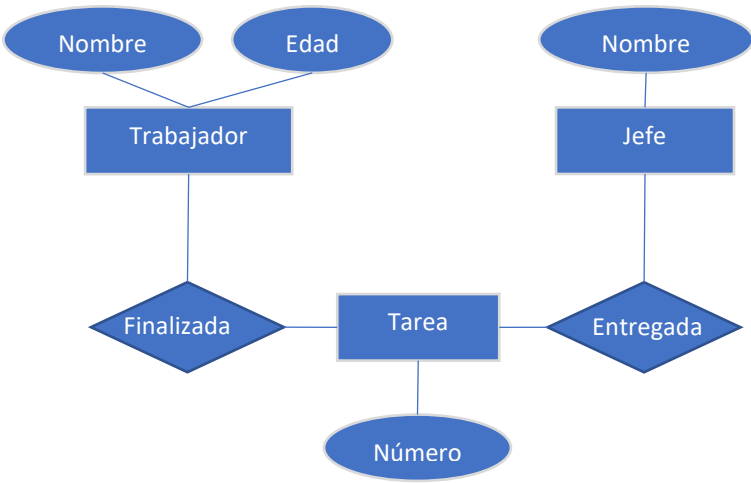
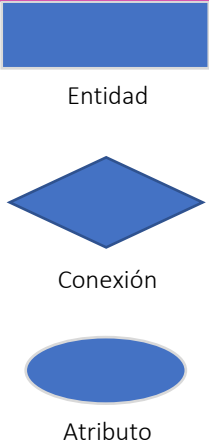
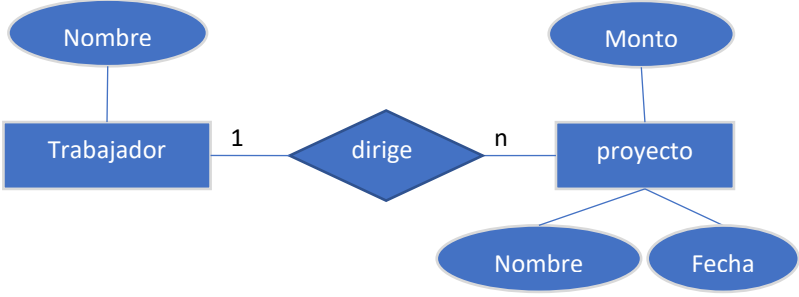
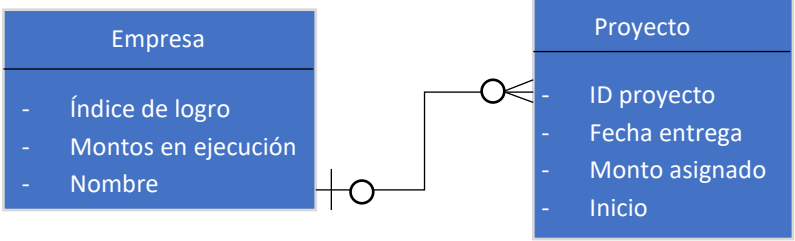
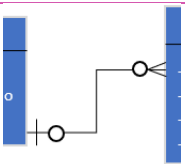
### Tiempo estimado

3 horas pedagógicas.

Desarrollo de la actividad

Situación experiencial

El docente presenta diferentes modelos entidad-relación y una progresión de la inclusión de símbolos para agregar más información:

Modelo entidad-relación	Descripción
	 <p>Entidad</p> <p>Conexión</p> <p>Atributo</p> <p>El jefe asigna tareas a los trabajadores y los trabajadores finalizan las tareas.</p>
	<p>1: n</p> <p>1 trabajador puede dirigir varios (n) proyectos</p>
	 <p>Relaciona dos entidades y expresa 1: n</p> <p>Cada proyecto puede ser ejecutado por una sola empresa. Así, se utiliza la llave primaria "Nombre" de la tabla "Banco" como una llave extranjera para muchos proyectos.</p>

Algunas de las preguntas que pueden motivar a los estudiantes a indicar las diferencias entre estos tres modelos de base de datos son:

- ¿Cuáles son las diferencias?
- ¿Qué es lo nuevo?
- ¿Qué semejanzas existen entre los modelos?
- ¿Para qué se puede usar cada una?

### Construcción de conocimiento

Para que los estudiantes sigan consolidando la habilidad de modelar una base de datos, se sugiere que el profesor les indique que la lectura cuidadosa de textos es esencial para inferir información que no esté explícita, es decir, el estudiante debería leer con atención para identificar las entidades, atributos y relaciones que están en el texto propuesto.

#### **Caso: Formulario de suscripción de subsidio**

*La empresa de subsidios Sub-TEX necesita implementar un sistema que permita gestionar los subsidios que tienen para vender. Las reglas del negocio de estos subsidios son explicadas por Simón, encargado de la puesta en marcha de estos subsidios. Él explica lo siguiente: cada año, entre febrero y marzo, las personas que ganan un sueldo líquido entre 600.000 a 1.000.000 de pesos pueden postular a subsidios en ciertos sectores de la ciudad, inscribiéndose en la oficina regional de Sub-TEX. Al inscribirse, el ciudadano debe señalar su Run, nombre completo, dirección, ciudad, correo electrónico, lugar donde trabaja (opcional) y su sueldo promedio del último año. Si el postulante es casado, además, debe señalar el nombre completo de su cónyuge. En cambio, si el postulante es soltero, debe señalar los años que lleva trabajando y su fecha de nacimiento. Un postulante debe presentarse como casado o soltero, pero no en ambas situaciones.*

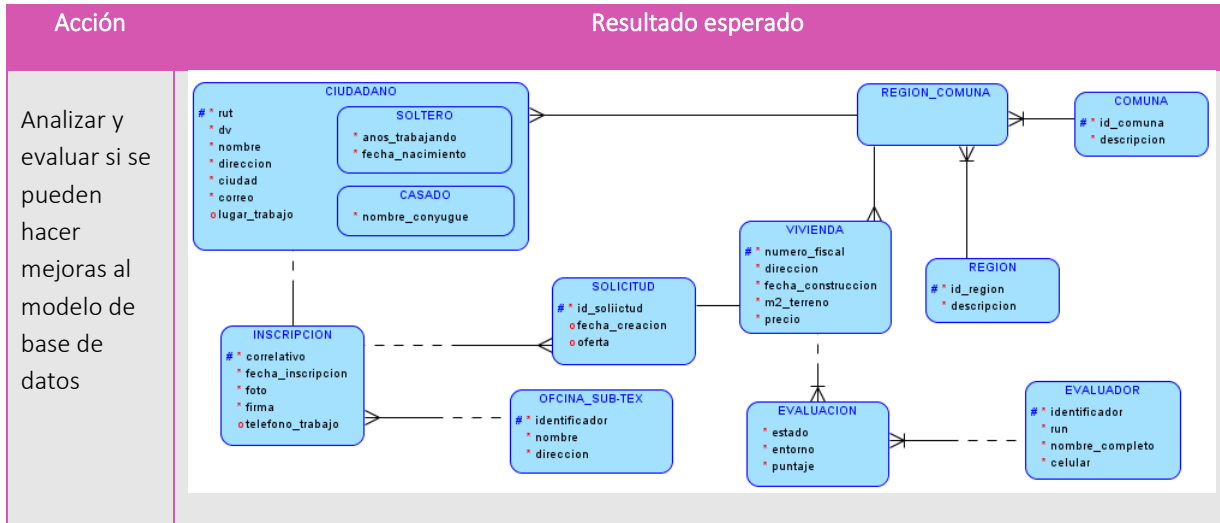
*Cada ciudadano puede inscribirse solo una vez en la vida. La inscripción debe señalar, además de todos los datos del ciudadano relacionado, el número correlativo único, la fecha de inscripción, foto del ciudadano y teléfono del lugar de trabajo (opcional).*

*Después de la inscripción, el ciudadano podrá solicitar un subsidio. En este documento se indican los datos de la vivienda que el ciudadano postulante quiere comprar. Estos son: identificador o número fiscal de vivienda, la dirección, fecha de construcción, metros cuadrados del terreno y precio de venta de la vivienda. Además, la solicitud registra el identificador de la solicitud, la fecha de creación y una contraoferta por la vivienda, que pueden ser opcionales. Cada postulante puede hacer tres solicitudes: una vivienda distinta en cada solicitud.*

*Los evaluadores de Sub-TEX deben “evaluar” el estado de cada vivienda inscrita. Cada evaluador debe hacer una evaluación de la vivienda. Esta evaluación contiene el estado de la vivienda, la descripción de la vivienda, las características de su entorno y le asigna un puntaje de 1 a 100. Cada solicitud de subsidio debe asociarse a una evaluación.*

Los evaluadores trabajan en parejas, por lo tanto, es necesario saber quién trabaja con quién. Además, es necesario saber el Run, nombre completo, teléfono celular y su respectivo identificador de evaluador.

Acción	Resultado esperado
<p>Marcar de color <b>AZUL</b> las entidades, de color <b>ROJO</b> los atributos obligatorios, de color <b>NARANJA</b> los atributos opcionales y de color <b>VERDE</b> los verbos que permitan reconocer relaciones entre entidades</p>	<p>Una empresa de subsidios “Sub-TEX” necesita implementar un sistema que permita gestionar los subsidios que tienen para vender. Las reglas del negocio de estos subsidios son explicadas por Félix Hojas, encargado de la puesta en marcha de estos subsidios. Él explica lo siguiente: cada año, entre febrero y marzo, las personas que ganan un sueldo líquido entre 600.000 a 1.000.000 de pesos pueden postular a subsidios en ciertos sectores de la ciudad, inscribiéndose en la oficina regional de Sub-TEX. Al inscribirse, el <b>ciudadano</b> debe señalar su <b>Run, nombre completo, dirección, ciudad, correo electrónico, lugar donde trabajar que es opcional, su sueldo promedio del último año</b>. Si el postulante es <b>casado</b>, además debe señalar el <b>nombre completo de su cónyuge</b>. En cambio, si el postulante es <b>soltero</b>, debe señalar los <b>años que lleva trabajando y su fecha de nacimiento</b>. Un postulante debe presentarse como casado o soltero, no ambas situaciones.</p> <p>Cada ciudadano puede inscribirse solo una vez en la vida. La <b>inscripción</b> debe señalar, además de todos los <b>datos del ciudadano relacionado, el número correlativo único, la fecha de inscripción, foto del ciudadano y teléfono del lugar de trabajo (opcionalmente)</b>.</p> <p>Después de la inscripción, el ciudadano podrá <b>solicitar un subsidio</b>. En este documento se indican los datos de la <b>vivienda</b> que el ciudadano postulante quiere comprar, estos son: <b>identificador o número fiscal de vivienda, la dirección, fecha de construcción, metros cuadrados del terreno y precio de venta de la vivienda</b>. Además, la <b>solicitud</b> registra el <b>identificador de la solicitud, la fecha de creación y una contraoferta por la vivienda, que pueden ser opcionales</b>. Cada postulante puede hacer tres solicitudes, una vivienda distinta en cada solicitud.</p> <p>Los evaluadores de Sub-TEX deben ir a evaluar el estado de cada vivienda inscrita. Cada evaluador debe hacer una <b>evaluación</b> de la vivienda. Esta evaluación contiene el <b>estado de la vivienda, la descripción de la vivienda, las características de su entorno y le asigna un puntaje de 1 a 100</b>. Cada solicitud de subsidio debe asociarse a una evaluación.</p> <p>Los <b>evaluadores</b> trabajan en parejas, por lo tanto, es necesario saber quién trabaja con quién. Además, es necesario saber el <b>Run, nombre completo, teléfono celular y su respectivo identificador de evaluador</b>.</p>
<p>Modelar la base de datos que acaba de clasificar en el problema</p>	<pre> classDiagram     class CIUDADANO {         rut         dv         nombre         direccion         ciudad         correo         lugar_trabajo     }     class SOLTERO {         anos_trabajando         fecha_nacimiento     }     class CASADO {         nombre_conyugue     }     class SOLICITUD {         id_solicitud         fecha_creacion         oferta     }     class OFICINA_SUB-TEX {         identificador         nombre         direccion     }     class EVALUADOR {         identificador         run         nombre_completo         celular     }     class VIVIENDA {         numero_fiscal         direccion         fecha_construccion         m2_terreno         precio     }     class EVALUACION {         estado         entorno         puntaje     }     CIUDADANO ..&gt; SOLTERO     CIUDADANO ..&gt; CASADO     CIUDADANO --&gt; INSCRIPCION     class INSCRIPCION {         correlativo         fecha_inscripcion         foto         firma         telefono_trabajo     }     INSCRIPCION ..&gt; CIUDADANO     INSCRIPCION ..&gt; OFICINA_SUB-TEX     INSCRIPCION --&gt; SOLICITUD     SOLICITUD --&gt; VIVIENDA     EVALUADOR ..&gt; EVALUACION     VIVIENDA ..&gt; EVALUACION     </pre>



### Práctica guiada

Para guiar la evaluación de modelar una base de datos con las nuevas relaciones, se recomienda tener claro que la mayor información debe obtenerse a través de la documentación, pero también la información puede venir mediante ejemplos de la interfaz que acompañan al modelo. A continuación, se muestran plantillas de diseños con casilleros para completar, o *mockups* de inicio de sesión, del formulario de registro y de selección de piezas básicas.

*En PC-Masters, un usuario común no puede iniciar sesión en esta empresa, ya que trabaja con empresas grandes a pedido. Por ello, el registro de usuarios solo lo puede hacer el administrador y nadie más.*

*Después de haber iniciado sesión, el usuario va a ser capaz de armar un PC con sus respectivas piezas, por ejemplo: placa madre, procesador, memoria RAM, etc. Una empresa puede armar muchos computadores, por ende, la empresa podrá seleccionar cuántos computadores planea obtener. Un usuario puede ser una empresa o un profesional, pero no ambos. A su vez, un profesional puede llegar a ser un administrador. Las piezas de computadores deben tener sus respectivas características. Al finalizar el pedido, deberá firmar digitalmente un contrato y esperar a que un profesional lo firme de vuelta y administre su pedido.*

**Iniciar sesion**

NOMBRE USUARIO

CONTRASEÑA

INICIAR SESION

---

**FORMULARIO DE REGISTRO**

RUT

CORREO ELECTRONICO

NOMBRE

APELLIDO

NOMBRE USUARIO

CONTRASEÑA

REGION

COMUNA

DIRECCION

TELEFONO

SELECCIONE SEXO

EMPRESA

PROFESIONAL

SELECCIONE EMPRESA

SELECCIONE SUCURSAL

FECHA CONTRATADO

SUBIR FOTO PERFIL

REGISTRARSE

---

**SELECCIONAR PIEZAS BASICAS**

Procesador (CPU) \*

(3985) Ryzen 5 - 5600X (AM4) \$300

AMD CPU RYZEN - 5600X (AM4)

ID: 3985 VER DETALLE \$300

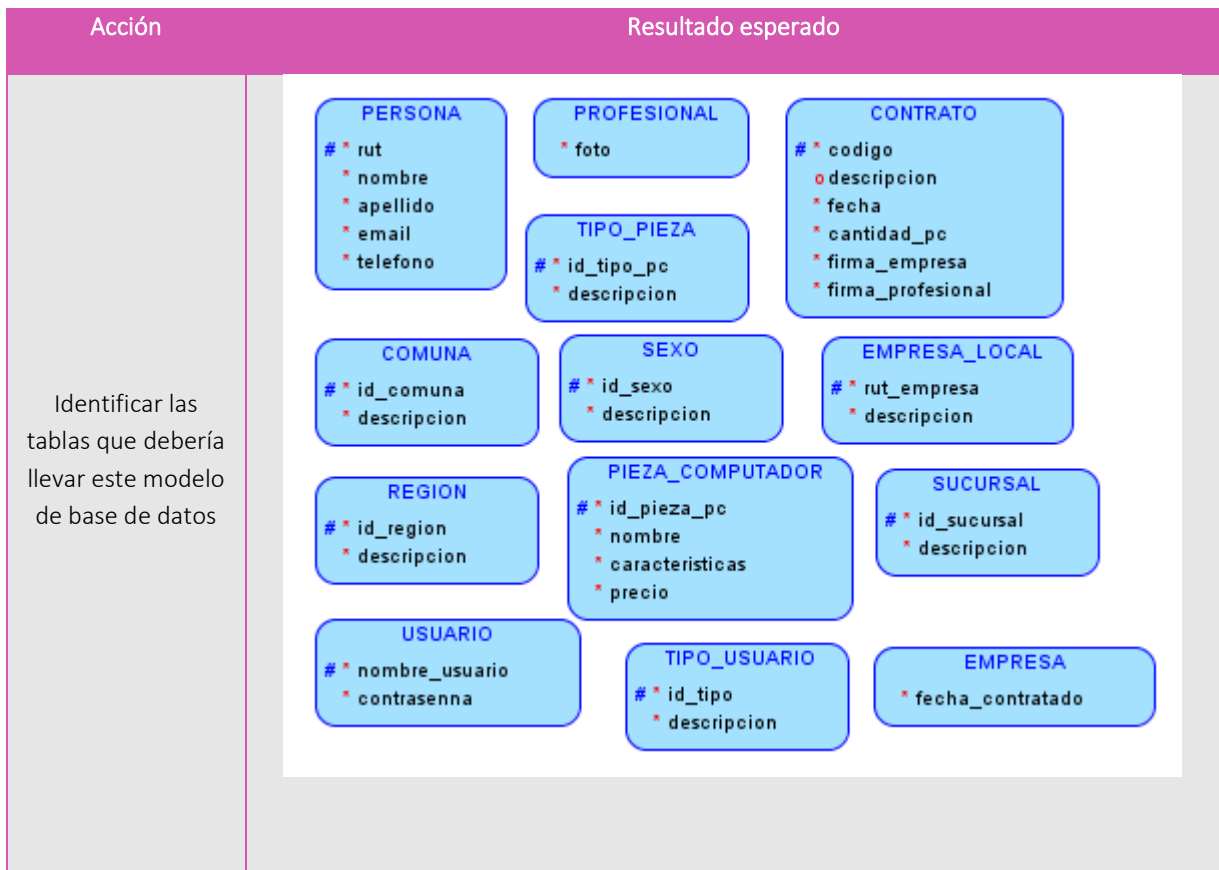
PLACA MADRE \*

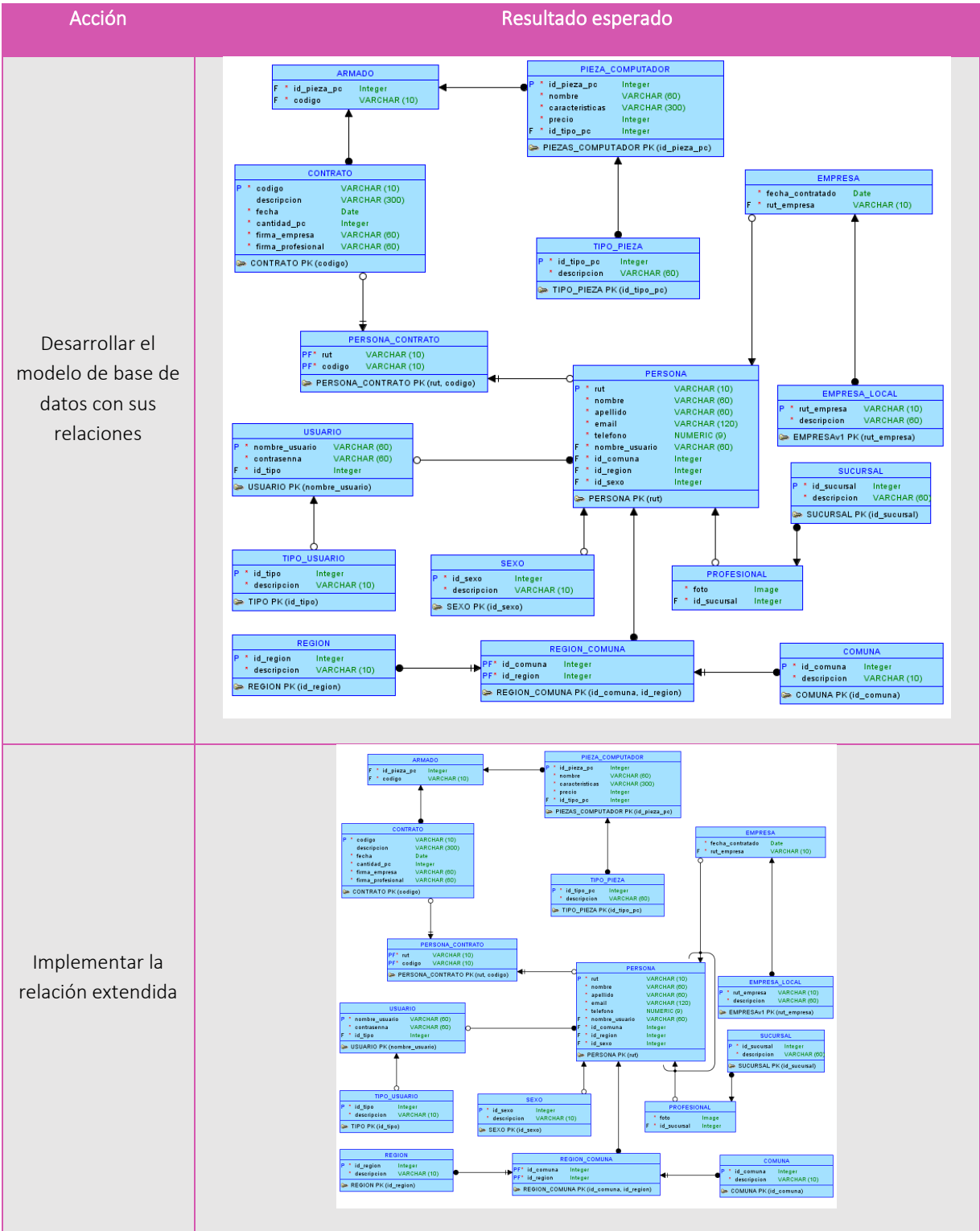
(4165) ASUS M/B AMD X570 TUG GAM

ASUS M/B AMD X570 TUF GAMING W1-FI (AM4)

ID: 4165 VER DETALLE \$200

A la hora de hacer modelos, se sugiere considerar los elementos que son factibles de conocer por todos, de tabular y que su significado sea directo; datos que no tengan alguna connotación diferente para el usuario, como lo es el uso de la clave “sexo”. Por otra parte, se acostumbra siempre a tener las mismas claves como “región” y “comuna”. Se sugiere promover diferentes claves según los trabajos o aspiraciones de la clase, teniendo presente que los productos, herramientas y pertenencias también tienen atributos que pueden ser consultados.

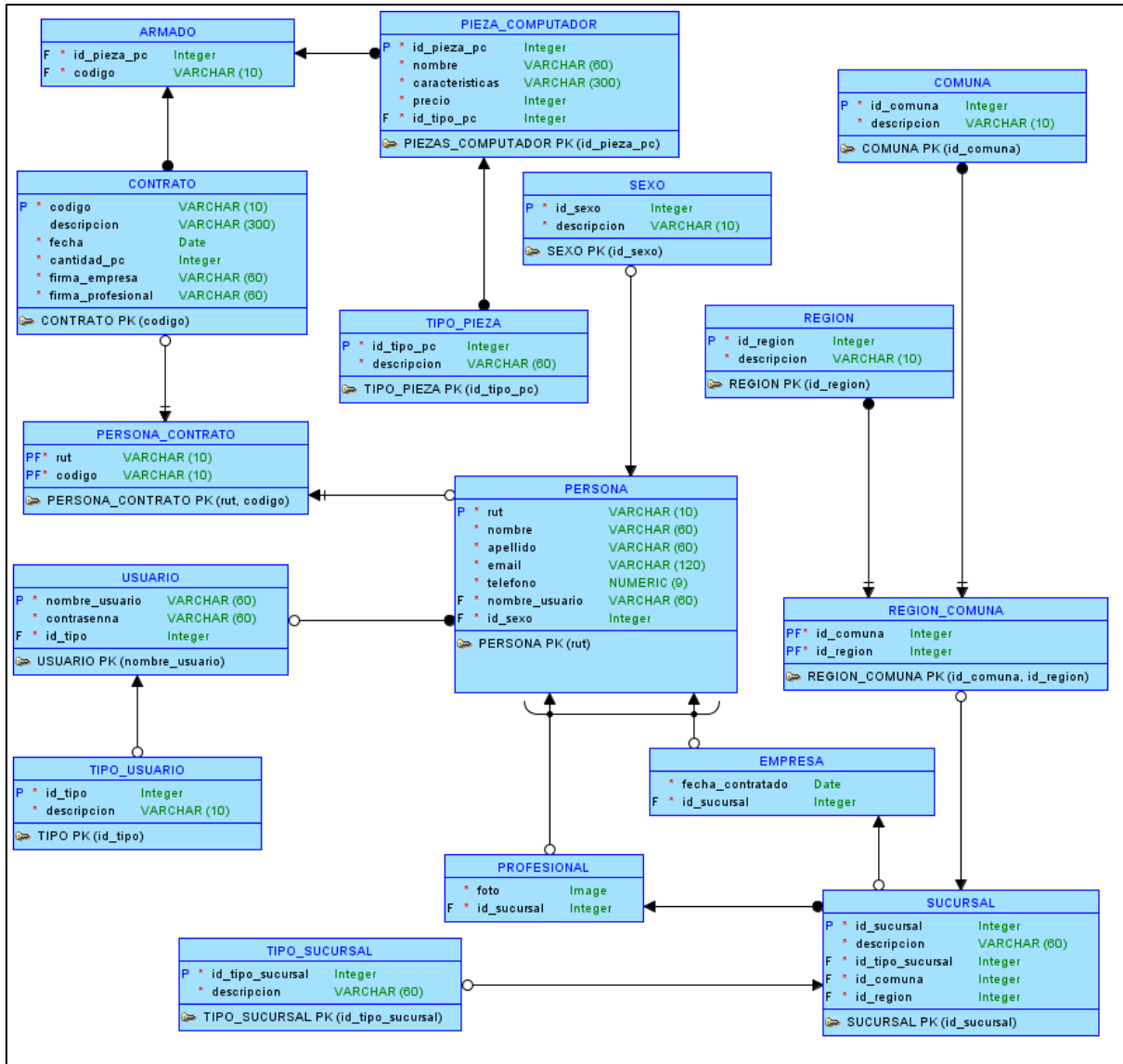






### Práctica independiente

Se sugiere una actividad individual de comparación de dos modelos entidad-relación. Para esto, se puede considerar el modelo obtenido en la práctica guiada y un modelo al cual se le han realizado algunos cambios. Se sugiere revisar punto por punto para ver en qué parte se hacen los cambios y cómo estos contribuyen a la optimización del modelo.



Relevar la selección de región y comuna junto con la entidad sucursal y observar que en vez de haber una sucursal y empresa local, se puede dejar solo la sucursal, como también las diferencias entre sucursales de la empresa o de las tiendas de PC-Masters.

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

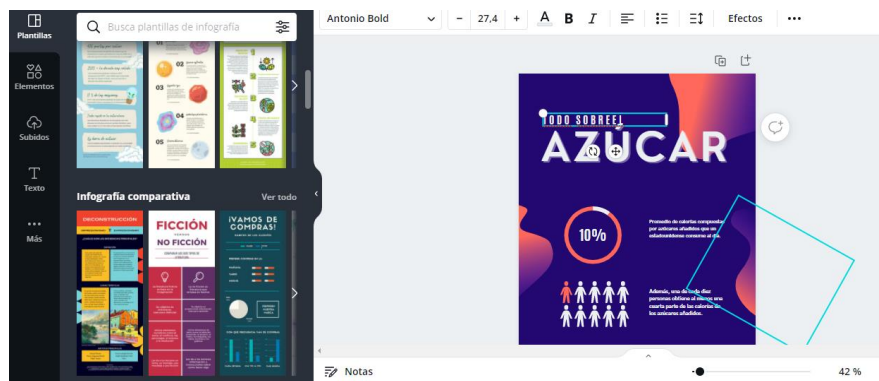
Criterio	Inicial	Intermedio	Avanzado
Problemática	Analiza el problema.	Analiza el problema y consigue obtener la mayoría de los datos.	Analiza el problema y logra identificar todos los datos de la información.
Tablas	Relaciona las tablas con errores.	Relaciona las tablas con algunos errores.	Relaciona las tablas sin errores.
Formas normales	Aplica las formas normales con errores.	Aplica las formas normales con algunos desaciertos.	Aplica las formas normales de buena manera, además de saber cuándo ya es suficiente.

### Orientaciones al docente

**Para unificar conceptos disciplinares:** Hay que tener en cuenta que cada aspecto de la base de datos cuenta para que el resultado final sea óptimo y funcione lo mejor posible a la hora de guardar información y complementar alguna aplicación por la cual se crea dicha base de datos. Siempre hay que priorizar que los estudiantes analicen cada aspecto de los problemas que se implementan en las actividades, ya que es fundamental tener una buena base de descomposición para así poder llegar a conseguir el resultado esperado. Los nuevos conceptos se pueden tomar de manera fácil, pero se debe tener en cuenta que estos conceptos son la 3ª forma normal. No obstante, muy pocas veces se llegan a utilizar normalmente, pero cuando se necesiten, allí estará esta nueva forma de implementar alternativas a la hora de resolver problemas.

- Los nuevos símbolos permiten explicar: jerarquías, herencia, exclusividad de relaciones y relaciones intransferibles.
- La jerarquía es el proceso de subdividir una entidad en varias subentidades, relacionándolas con la entidad a la que se refieren o a la entidad PADRE, por lo tanto, las subentidades son HIJOS.
- La herencia, por su parte, es la implementación de la jerarquía de subtipos y supertipos, dado que los subtipos heredan todos los atributos de la entidad supertipo.
- El arco exclusivo es una restricción sobre dos o más relaciones de una entidad. Un arco indica que cualquier instancia de una entidad puede tener solo una relación válida respecto a las relaciones que están rodeadas por el arco. Por lo tanto, el arco representa una relación exclusiva de las relaciones entre el arco y puede usarse también en relaciones de subtipo.
- Reglas del subtipo:
  - Un subtipo hereda todos los atributos del supertipo.
  - Un subtipo hereda todas las relaciones del supertipo.
  - Un subtipo normalmente tiene sus propios atributos o relaciones.
  - Un subtipo nunca existe sólo, requiere del supertipo.
  - Un subtipo también se conoce como subentidad.
  - Un subtipo puede tener sus propios subtipos, estos se denominan subtipos anidados.

- Un subtipo hereda todos los atributos del supertipo.
  - Un subtipo hereda todas las relaciones del supertipo.
  - Un subtipo normalmente tiene sus propios atributos o relaciones.
  - Un subtipo nunca existe sólo, requiere del supertipo.
  - Un subtipo también se conoce como subentidad.
  - Un subtipo puede tener sus propios subtipos, estos se denominan subtipos anidados.
- La aplicación utilizada para el modelamiento de base de datos es Oracle DataModeler<sup>20</sup>.
  - Un *mockup* es una plantilla que viene con un diseño y sirve para replicar u obtener información. En estas plantillas se tienen casilleros que pueden ser cambiados con nueva información. Se pueden encontrar plantillas de libre disposición en <https://link.curriculumnacional.cl/www.canva.com> y se pueden utilizar plantillas manuales en caso de no tener acceso a internet. Estas se elaboran con cartón y espacios en este. En el caso manual, se sugiere practicar el tipo de letra o tener plantillas con tipos de letras.



**Actitudes:** Para apoyar el desarrollo de las actividades, es recomendable el uso de los computadores a la hora de modelar la base de datos, para que así obtengan una retroalimentación más visual y puedan ver los aciertos y desaciertos que logran a la hora de resolver los problemas. En cuanto al desarrollo del trabajo individual, se sugiere promover la independencia y pensamiento propio del estudiante.

**Orientaciones para organizar e implementar el juego de roles:** Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Elaborar un ejercicio amplio que esté centrado en algún sistema de negocio o minimercado, para que los alumnos identifiquen las tablas necesarias. Se sugiere pensar también en las relaciones que se dan la situación para luego utilizarlas en las próximas actividades.
- Determinar un tiempo para poder analizar y revisar todo el problema; para identificar las tablas que esconde.

<sup>20</sup> Posible descargar desde:  
<https://curriculumnacional.cl/link/https://www.oracle.com/cl/database/technologies/appdev/datamodeler.html>

- Llamar a los estudiantes a revisar entre todas las respuestas que escogieron. Esto beneficiará a los que no lograron identificarlas todas, además de poder aclarar dudas de cómo identificar cada caso. Una vez finalizada la retroalimentación con los estudiantes, deberán trabajar en las mejoras del modelamiento de la situación e incluir aquellas relaciones o pasos que son necesarios. Puede ser en una hoja, pero se recomienda siempre el uso de computadores para estos casos.
- Determinar el tiempo para poder desarrollar el modelo de base de datos con sus respectivas entidades, atributos, relaciones y aplicar las formas normales.
- Dar retroalimentación de la tarea llamando a algún estudiante o simplemente mostrándoles al final de la clase cómo debería ser el modelo y por qué debería quedar así. Explicar cómo lo analizó y desarrolló.

## Actividad de desempeño 2

### Propósito

En esta actividad se retomará el funcionamiento del algoritmo incorporando una nueva función de la estructura de los repetidores o ciclos con el fin de concluir esta estructura y, a su vez, fortalecer el conocimiento de esta con más práctica, al tiempo que se favorece analizar los problemas y así dar las mejores soluciones posibles.

### Objetivos de Aprendizaje

**OA2.** Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función.  
**(Evaluar)**

### Conocimiento esencial

Sistemas computacionales: dispositivos, *hardware* y *software*, y resolución de problemas.

### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

El docente presenta una situación cotidiana como ir a un cajero automático a sacar dinero, tener que poner la clave y suponer que el usuario no se acuerda de la clave.



Se sugiere conversar en términos de la programación lo que ocurre en esta situación. Algunas preguntas que pueden fomentar las nociones de las funciones **mientras** y **para** son:

- ¿Cuáles son las preguntas o instrucciones del cajero automático?
- ¿Qué ocurre si nos olvidamos de la clave?
- ¿Qué consideraciones hay que tener para elaborar un algoritmo?

### Construcción de conocimiento

Teniendo en cuenta estas nuevas formas de implementar, se requiere de un código capaz de funcionar como cajero automático, donde el usuario deba ingresar una contraseña y si se equivoca 3 veces, su tarjeta se bloquea y finaliza la tarea.

El menú tendrá las siguientes opciones:

```
***** CAJERO AUTOMÁTICO *****
*                               *
*           INTENTO NRO 1       *
*                               *
*****
CLAVE : > |
```

```
MENU DE OPCIONES
1. VER SALDO
2. DEPOSITO
3. RETIRO
4. SALIR
OPCIÓN : >
```

- **Ver saldo:** uno podrá revisar el saldo que tiene (puede o no puede tener un saldo prefijado).
- **Depósito:** uno podrá hacer un depósito a su cuenta.
- **Retiro:** Se retirará el dinero que tenga como saldo (si no tiene el saldo suficiente, no podrá retirar el dinero).

Acción	Resultado esperado	Preguntas de apoyo
<p>Realizar el menú de las opciones del cajero automático</p>	<pre> <b>Definir</b> saldo, monto <b>como</b> Real; <b>Definir</b> contador, opc, salir <b>como</b> Entero; <b>Definir</b> usuario, clave <b>como</b> Caracter; //SALDO DEL CLIENTE saldo ← 100000; // CLAVES DE ACCESO usuario ← '5555';  <b>Mientras</b> (contador ≠ 4 ) <b>Hacer</b>     <b>Si</b> (contador ≠ 3 ) <b>Entonces</b>         <b>Escribir</b> "***** CAJERO AUTOMÁTICO *****";         <b>Escribir</b> "*          INTENTO NRO ", contador + 1, "          *";         <b>Escribir</b> "*****";         <b>Escribir</b> " ";         <b>Escribir Sin Saltar</b> "CLAVE : ";         <b>Leer</b> clave;          <b>Escribir</b> "";         <b>Limpiar Pantalla</b>;         <b>Si</b> (clave == usuario) <b>Entonces</b>             <b>Mientras</b> (salir ≠ 4) <b>Hacer</b>                 <b>Limpiar Pantalla</b>;                 <b>Escribir</b> "* BIENVENIDOS AL BANCO *";                 <b>Escribir</b> "MENU DE OPCIONES";                 <b>Escribir</b> "1. VER SALDO";                 <b>Escribir</b> "2. DEPOSITO";                 <b>Escribir</b> "3. RETIRO";                 <b>Escribir</b> "4. SALIR";                 <b>Escribir Sin Saltar</b> "OPCIÓN : ";                 <b>Leer</b> opc;             </pre>	<p>¿Cómo verificar si el usuario es correcto o incorrecto?</p> <p>¿Qué ciclo o repetidor usar?</p>
<p>Realizar las operaciones que hará el cajero automático</p>	<pre> <b>Segun</b> (opc) <b>Hacer</b>     1: <b>Escribir</b> "SALDO ACTUAL : ", saldo;        <b>Esperar Tecla</b>;      2: <b>Escribir Sin Saltar</b> "MONTO A DEPOSITAR : ";        <b>Leer</b> monto;        saldo ← saldo + monto;      3: <b>Escribir Sin Saltar</b> "MONTO A RETIRAR : ";        <b>Leer</b> monto;         <b>Si</b> (monto &gt; saldo) <b>Entonces</b>            <b>Escribir</b> "¡Error no dispone de ese Saldo!..";            <b>Esperar Tecla</b>;        <b>SiNo</b>            saldo ← saldo - monto;        <b>FinSi</b>      4: <b>Escribir</b> "Gracias por su visita, regrese pronto...";        salir ← 4;        contador ← 4;  <b>FinSegun</b> <b>FinMientras</b> </pre>	

Acción	Resultado esperado	Preguntas de apoyo
Finalizar las operaciones del cajero automático	<pre> SiNo     contador ← contador + 1; FinSi  SiNo     Escribir "*****";     Escribir "*                *";     Escribir "*   TARJETA BLOQUEADA   *";     Escribir "*                *";     Escribir "*****";     contador ← 4; FinSi  FinMientras         </pre>	

### Práctica guiada

Recordando la actividad de PC-Masters, ahora los estudiantes asumirán la perspectiva del programador. La forma de hacerlo no es la mejor, pero se debería entender que debido a que no se está trabajando con una base de datos y mucho menos con dimensiones, hay limitaciones a la hora de crear esto en un pseudocódigo.

Acción	Resultado esperado
Realizar el menú.	<pre> Algoritmo PCMasters     Repetir         Limpiar Pantalla;         Escribir "===== ";         Escribir "  BIENVENIDOS AL PC-Masters  ";         Escribir "===== ";         Escribir "";         Escribir "MENU DE OPCIONES";         Escribir "1. Identificar usuario";         Escribir "2. Pedir armado de pc";         Escribir "3. Firmar contrato";         Escribir "4. SALIR";         Escribir "";         Escribir Sin Saltar "OPCIÓN : ";         Leer aux;         Escribir "";     </pre>



Acción	Resultado esperado
<p>Confeccionar las opciones dentro de un "según".</p>	<pre>Segun aux 1: Escribir Sin Saltar "Ingrese su nombre: " Leer Nombre;</pre>
<p>Seguir con la siguiente opción y confeccionar un guardado de información más explícito.</p>	<pre>2: Escribir "SELECCIONE PROCESADOR"; Escribir "1. Intel core i5 11600K \$259.990"; Escribir "2. Intel core i9 10900k \$519.990"; Escribir "3. AMD Ryzen 5 5600x \$279.990"; Escribir "4. AMD Ryzen 9 5950x \$799.990"; Leer aux; Segun aux 1: Procesador = "Intel core i5 11600K"; ProcesadorV = 259990; 2: Procesador = "Intel core i9 10900k"; ProcesadorV = 519990; 3: Procesador = "AMD Ryzen 5 5600x"; ProcesadorV = 279990; 4: Procesador = "AMD Ryzen 9 5950x"; ProcesadorV = 799990; FinSegun PrecioTotal = PrecioTotal + ProcesadorV;</pre>

Acción	Resultado esperado
<p>Dependiendo de la opción que elija, hacer opciones específicas para cada usuario (Intel o AMD)</p> <p>Hay que resaltar que también deberá existir una opción para la tarjeta gráfica, memoria RAM, disco duro, gabinete y fuente de poder.</p>	<pre> Si aux = 1 o aux = 2 Entonces   Escribir "SELECCIONE PLACA MADRE";   Escribir "1. Intel - ASUS ROG STRIX B365-F GAMING \$159.990";   Escribir "2. Intel - Gigabyte Z390 AORUS ELITE \$179.990";   Leer aux;   Segun aux     1:       PlacaMadre = "Intel - ASUS ROG STRIX B365-F GAMING";       PlacaMadreV = 159990;     2:       PlacaMadre = "Intel - Gigabyte Z390 AORUS ELITE";       PlacaMadreV = 179990;   FinSegun SiNo   Escribir "SELECCIONE PLACA MADRE";   Escribir "1. AMD - Gigabyte B450 I AORUS PRO WIFI \$139.990";   Escribir "2. AMD - ASUS ROG Crosshair VIII Dark Hero WI-FI \$459.990";   Leer aux;   Segun aux     1:       PlacaMadre = "AMD - Gigabyte B450 I AORUS PRO WIFI";       PlacaMadreV = 139990;     2:       PlacaMadre = "AMD - ASUS ROG Crosshair VIII Dark Hero WI-FI";       PlacaMadreV = 459990;   FinSegun FinSi PrecioTotal = PrecioTotal + PlacaMadreV; </pre>
<p>Finalizar con la última opción y dar un resumen de lo seleccionado anteriormente.</p> <p>Preguntar cuántos computadores serán y luego confirmar la orden.</p>	<pre> 3:   Escribir "PIEZAS DE COMPUTADOR";   Escribir "Nombre del comprador: ", Nombre;   Escribir "Procesador: ", Procesador, " - \$", ProcesadorV;   Escribir "Placa Madre: ", PlacaMadre, " - \$", PlacaMadreV;   Escribir "Tarjeta Video: ", TarjetaGrafica, " - \$", TarjetaGraficaV;   Escribir "Memoria RAM: ", MemoriaRam, " - \$", MemoriaRamV;   Escribir "Disco Duro: ", DiscoDuro, " - \$", DiscoDuroV;   Escribir "Gabinete: ", Gabinete, " - \$", GabineteV;   Escribir "Fuente de Poder: ", FuentePoder, " - \$", FuentePoderV;   Escribir "Precio TOTAL: \$", PrecioTotal;   Repetir     Escribir "Cuántos computadores serán?"     Leer cant;     PrecioTotalTotal = PrecioTotal * cant;      Escribir "Precio total por la cantidad de ",cant," computadores: \$",PrecioTotalTotal;     Escribir "Desea confirmar la orden?"     Escribir "1. Confirmar";     Escribir "2. Desconfirmar";     Escribir "";     Escribir Sin Saltar "OPCIÓN : ";     Leer aux;   Hasta Que aux = 1   FinSegun Hasta Que aux = 4 FinAlgoritmo </pre>

### Práctica independiente

Se sugiere que el docente proponga un problema que se resuelva individualmente. Los problemas deben ser hechos específicamente para utilizar algún ciclo o repetidor. Por ejemplo: en un nuevo videojuego donde se enfrenta un jugador contra un robot, se necesita determinar quién gana el juego. Cuando el juego comienza, el jugador se enfrenta al robot y ambos tendrán 50 energía de daño. El jugador y el robot comienzan con 500 de energía. El jugador se enfrenta al robot y se determina cuál de los dos golpea primero mediante un número al azar; el que golpea primero gana energía.

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Seudocódigo	Elabora un pseudocódigo.	Elabora un pseudocódigo que corresponde medianamente a lo solicitado.	Elabora un pseudocódigo que corresponde a lo solicitado.
Condicionales	Elabora las estructuras condicionales con errores.	Elabora las estructuras condicionales con algunos errores.	Elabora las estructuras condicionales sin ningún error.
Repetidores o ciclos	Elabora mal las estructuras cíclicas.	Elabora de manera moderada las estructuras cíclicas.	Aplica las estructuras cíclicas de manera correcta y sin errores.
Código	Elabora un código ordenado.	Elabora un código ordenado y medianamente funcional (falta corregir detalles).	Elabora un código ordenado y funcional (funciona en cada momento de su ejecución).

### Orientaciones al docente

**Para unificar conceptos disciplinares:** Entendemos que el algoritmo es algo más que instrucciones que siguen un orden. Con las nuevas herramientas esto termina teniendo más de una dirección a la hora de trabajarla. Por esto, la función “mientras” nos permite llevar a cabo algoritmos más complejos y cercanos a la realidad, además de complementar la categoría de los ciclos y funciona de la siguiente manera:

- Mientras: Este ciclo nos permite repetir mientras se cumple una acción en concreto, se podría decir que este ciclo en particular es la contraparte del ya mencionado “repetir”.

Con la estructura condicional los algoritmos seguían siendo lineales, pero con la estructura de ciclos se abren nuevas brechas que nos permiten repetir en más de una ocasión ciertas actividades que son cruciales a la hora de resolver los problemas que vienen en el futuro.

Se sugiere una tabla para organizar los conocimientos nuevos con los anteriores y así comparar una misma situación con diferentes funciones de programación, donde uno funciona con “mientras” y el otro con “para”, por ejemplo:

**Función: Mientras**

Elaborar un programa que indique, después de haber ingresado todas las notas, el promedio de las notas

<p>Programa promedio de notas</p> <p>Variable <math>i</math> contador de notas <math>i</math>: número entero</p> <p>Variable <math>s</math> suma de notas <math>s</math>: número racional</p> <p>Variable <math>p</math> promedio de notas <math>p</math>: número racional</p> <p><math>i \leftarrow 0</math></p> <p><math>s \leftarrow 0</math></p> <p><b>Mientras <math>i \leq 10</math></b></p> <p>Entrar número <math>s</math></p> <p>Leer número <math>s</math></p> <p>Suma <math>s := s + s</math></p> <p><math>i \leftarrow i + 1</math></p> <p><b>Fin Mientras</b></p> <p>Escribir: “Ya ingresaron 10 notas”</p> <p>Promedio: <math>p = \frac{\text{suma}}{i}</math></p> <p>Escribir: “El promedio es <math>p</math>”</p> <p>Fin Programa</p>	<pre> graph TD     Start([Algoritmo Mientras01]) --&gt; Input1[/Ingrese la cantidad de Notas :/]     Input1 --&gt; Assign1[Num_Notas]     Assign1 --&gt; Decision1{cont &lt; Num_Notas}     Decision1 -- V --&gt; Assign2[cont + cont + 1]     Assign2 --&gt; Input2[/Ingrese Nota ' , cont, ' de ' , Num_Notas, ' : '/]     Input2 --&gt; Assign3[Nota]     Assign3 --&gt; Decision2{Nota &gt; 40}     Decision2 -- F --&gt; Assign4[Nota_Des + Nota_Des + Nota]     Decision2 -- V --&gt; Assign5[Nota_Apro + Nota_Apro + Nota]     Assign4 --&gt; Assign6[Pro_Des + Pro_Des + 1]     Assign5 --&gt; Assign7[Pro_Apro + Pro_Apro + 1]     Assign6 --&gt; Decision1     Assign7 --&gt; Decision1     Decision1 -- F --&gt; Output1[/Promedio de Notas Aprobadas : ' , Nota_Apro / Pro_Apro /]     Output1 --&gt; Output2[/Promedio de Notas Desaprobadas : ' , Nota_Des / Pro_Des /]     Output2 --&gt; End([FinAlgoritmo])     </pre>
---	---

**Función: Para**

Calcular la suma de los números naturales desde 1 hasta n

Inicio

Variable  $i$  : número entero

Variable  $n$  : número entero

Variable  $s$  suma : número entero

Leer ( $n$ )

$i = 1$

**Para  $i = 1$  hasta  $n$  hacer**

Suma = suma +  $i$

**Fin\_Para**

Escribir: Suma

Fin Programa

```

graph TD
    Start([Algoritmo Para01]) --> Input1[/Ingrese la cantidad de Notas : /]
    Input1 --> NumNotas[Num_Notas]
    NumNotas --> Input2[/Ingrese Nota 'cont,' de 'Num_Notas,' : /]
    Input2 --> Nota[Nota]
    Nota --> Decision{Nota > 40}
    Decision -- F --> CalcDes[Nota_Des ← Nota_Des + Nota  
Pro_Des ← Pro_Des + 1]
    Decision -- V --> CalcApro[Nota_Apro ← Nota_Apro + Nota  
Pro_Apro ← Pro_Apro + 1]
    CalcDes --> Loop((cont | Num_Notas))
    CalcApro --> Loop
    Loop --> Output1[/Promedio de Notas Aprobadas : ',Nota_Apro/Pro_Apro/ /]
    Loop --> Output2[/Promedio de Notas Desaprobadas : ',Nota_Des/Pro_Des/ /]
    Output1 --> End([FinAlgoritmo])
    Output2 --> End
    
```

Las delimitaciones de “para” y “hasta” se pueden hacer en relación con otras variables, por ejemplo con “k” y “j”, indicando la instrucción “Para  $i = k$  hasta  $i = j$ ” que permite programar estas delimitaciones.

**Actitudes:** Para apoyar el desarrollo de las actividades es recomendable utilizar los computadores, ya que los problemas que vienen ahora terminan siendo más complejos y a la vez más largos y es necesario tener una retroalimentación en el momento, para así poder tener más confianza de lo que uno llega a programar. Por otra parte, en cuanto al desarrollo de las actividades individuales es necesario tener claro que esto es para que el estudiante se sienta en confianza de sus conocimientos y en su forma de pensar.

**Orientaciones para organizar e implementar la actividad:** Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual del videojuego:

- Determinar un tiempo para así resolver cualquier duda del problema.
- Determinar el tiempo que cree es necesario para haber terminado o al menos llevar algo de desarrollo del ejercicio.
- Determinar el tiempo para retroalimentar a los estudiantes y guiarlos de acuerdo con las necesidades que vayan presentando.
- Favorecer que algún estudiante presente su resultado; puede ser por voluntad propia o elegido por el docente. Esto para poder dar una retroalimentación o refuerzo si está todo correcto. Además de aclarar dudas finales acerca del desarrollo de la actividad y poner un nuevo reto, ahora quizás el videojuego en vez de ser de solo un jugador sea entre dos.

## Actividad de desempeño 3

### Propósito

En esta actividad, se pone énfasis en el uso de los subprocesos y el cómo estos facilitan la organización del código, esto sirve para que los estudiantes tengan clara la transferencia de datos/información entre proceso y proceso. También, se reforzará lo aprendido anteriormente en los contextos de nuevos conocimientos, ya que en todo momento siempre se termina utilizando todas las metodologías que se van aprendiendo progresivamente en los procesos.

### Objetivos de Aprendizaje

**OA2.** Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función.  
**(Evaluar)**

**OA3.** Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles.  
**(Modelar)**

### Conocimiento esencial

- Modelos entidad-relación y simulación de sistemas.
- Sistemas computacionales: dispositivos, Hardware y Software y resolución de problemas

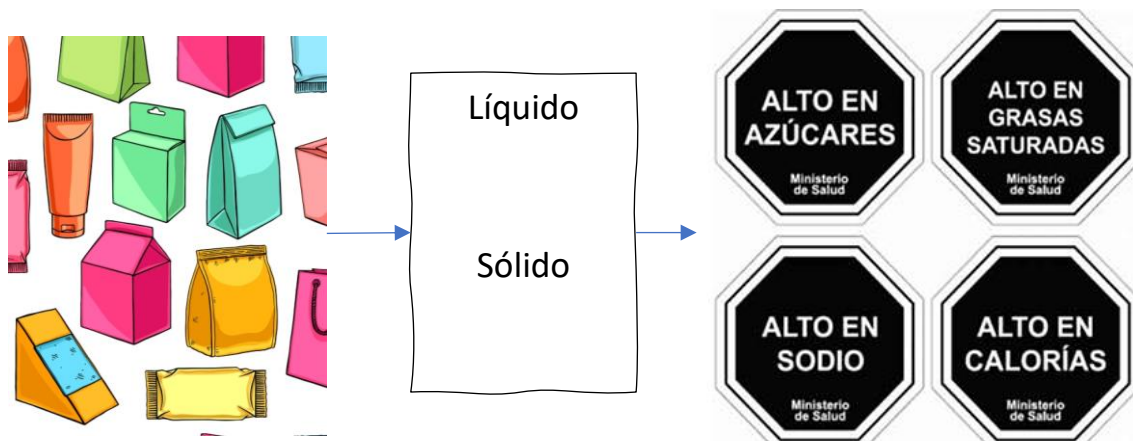
### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

El docente presenta una situación de clasificación de productos donde se requiere hacer primero una clasificación del estado de la materia (líquido o sólido) y a partir de esta clasificación se dan condiciones para asignar el logo correspondiente de alto en azúcares, alto en grasas saturadas, alto en sodio o alto en calorías.



Para motivar la descomposición de la programación en partes o en subprocesos se sugieren las siguientes preguntas:

- ¿Cómo describirías el proceso de asignación de logos a un producto alimenticio?
- ¿Qué partes tiene el proceso?
- ¿Cómo podría ser la programación completa?

### Construcción de conocimiento

Para construir el conocimiento, se sugiere comenzar con la metodología del subproceso y el problema de etiquetar los alimentos con las etiquetas definidas por la ley de alimentos. El alimento puede ser sólido o líquido y dependiendo del tipo estos tienen un valor límite para las 4 etiquetas que existen: alto en azúcares, alto en grasas saturadas, alto en sodio y alto en calorías.

Tabla N1 Límites de alimentos sólidos		Tabla N1 Límites de alimentos líquidos	
Nutriente o energía	Datos	Nutriente o energía	Datos
Azúcares totales g/100g	20	Azúcares totales g/100g	2
Grasas saturadas g/100g	5	Grasas saturadas g/100g	4
Sodio mg/100g	700	Sodio mg/100g	80
Energía kcal/100g	300	Energía kcal/100g	80



La actividad se desarrolla con los subprocesos presentes.

Acción	Resultado esperado	Preguntas de apoyo
<p>Elaborar primeros pasos del código.</p>	<pre> SubProceso SolicitarDatos(menu)  FinSubProceso Algoritmo Etiquetas   Definir opc Como Entero;    Repetir     Escribir "Ingrese tipo de alimento";     Escribir "1) Solido";     Escribir "2) Liquido";     Escribir "3) Salir";     Leer opc;      Segun opc       1: SolicitarDatos(1);       2: SolicitarDatos(2);       3: Escribir "Vuelva pronto!";     De Otro Modo:       Escribir "Ingrese opcion de menu correcta";     FinSegun   Hasta Que opc = 3 FinAlgoritmo           </pre>	<p>¿Qué se espera del subproceso?</p>
<p>Elaborar dentro del subproceso el ingreso de los datos que se compararán.</p>		



Acción	Resultado esperado	Preguntas de apoyo
	<pre> SubProceso SolicitarDatos(menu)    Definir calorías,sodio,grasas,azucar Como Entero    Escribir "Ingrese Azucares totales g/100g";   Leer azucar;   Escribir "Ingrese Grasas saturadas g/100g";   Leer grasas;   Escribir "Ingrese Sodio mg/100g";   Leer sodio;   Escribir "Ingrese Energia kcal/100g";   Leer calorías;           </pre>	
<p>Dependiendo de la opción elegida, comparar los datos ingresados y finalizar el código.</p>	<pre> Segun menu   1:     Si azucar &gt; 20 Entonces       Escribir "Alto en azucares totales";     FinSi     Si grasas &gt; 5 Entonces       Escribir "Alto en grasas saturadas";     FinSi     Si sodio &gt; 700 Entonces       Escribir "Alto en sodio";     FinSi     Si calorías &gt; 300 Entonces       Escribir "Alto en calorías";     FinSi   2:     Si azucar &gt; 2 Entonces       Escribir "Alto en azucares totales";     FinSi     Si grasas &gt; 4 Entonces       Escribir "Alto en grasas saturadas";     FinSi     Si sodio &gt; 80 Entonces       Escribir "Alto en sodio";     FinSi     Si calorías &gt; 80 Entonces       Escribir "Alto en calorías";     FinSi   FinSegun FinSubProceso           </pre>	

### Práctica guiada

Se sugiere proponer a los estudiantes la siguiente situación: Una clínica requiere para el área de nutrición un programa que permita realizar un cálculo del índice de masa corporal. Para esto se requiere la información de peso en kilogramos (kg) y de altura en metros (m).

La fórmula para sacar el cálculo es el siguiente:

$$IMC = \frac{\text{Peso (kg)}}{\text{Altura}^2 \text{ (m)}}$$

Diagnóstico	IMC
PESO INSUFICIENTE	<18.5
PESO NORMAL	18.5 – 24.9
SOBREPESO	25 – 29.9
OBESIDAD GRADO 1	30 – 34.9
OBESIDAD GRADO 2	35 – 39.9
OBESIDAD GRADO 3	≥ 40

Se requiere obtener la siguiente información:

- Categorización por paciente.
- La cantidad de pacientes por categoría.
- La categoría que tiene mayor cantidad de pacientes.

Para cada requerimiento se requiere su propio procedimiento.

Acción	Resultado esperado
<p>Crear un algoritmo listo para incorporar los tres subprocesos que tendrá la resolución del problema.</p> <p>No es necesario crearlas todas de una vez; se puede trabajar primero desde el proceso mismo y después pasarlo a un subproceso.</p>	<pre> <b>Proceso</b> ProcesoIMC  <b>Definir</b> cantidadClientes, i, ipe,pno,sg,ob1,ob2,ob3 <b>Como Entero</b>; <b>Definir</b> peso, talla <b>Como Real</b>;  <b>Escribir</b> "Bienvenido a IMC"; <b>Escribir</b> "Indique la cantidad de clientes"; <b>Leer</b> cantidadClientes;  <b>Para</b> i = 1 <b>Hasta</b> cantidadClientes <b>Hacer</b>     <b>Escribir</b> "-----";     <b>Escribir</b> "Ingrese peso del cliente N°",i;     <b>Leer</b> peso;     <b>Escribir</b> "Ingrese altura en metros del cliente N°",i;     <b>Leer</b> talla;      <b>Escribir</b> "El cliente ingresado esta en la categoria:";     <b>Escribir</b> "-----";     categoriaIMC(peso,talla,ipe,pno,sg,ob1,ob2,ob3);     //Llamar otro subproceso para que cuente la categoria. <b>FinPara</b>     <b>Escribir</b> "-----";     <b>Escribir</b> "Cantidad de clientes por categoria:";     <b>Escribir</b> "-----";     cantCliente(ipe,pno,sg,ob1,ob2,ob3);     <b>Escribir</b> "-----";     <b>Escribir</b> "La categoria con mas clientes es:";     <b>Escribir</b> "-----";     cantIMC(ipe,pno,sg,ob1,ob2,ob3);     <b>Escribir</b> "-----";  <b>FinProceso</b>         </pre>

Acción	Resultado esperado
<p>Implementar el primer subproceso donde se hará el cálculo y la respectiva cuenta de los pacientes para cada categoría.</p>	<pre> <b>SubProceso</b> categoriaIMC(peso,talla,ipe <b>Por Referencia</b> <b>Definir</b> imc <b>Como Real</b>;  imc = peso / talla <math>\uparrow</math> 2;  <b>Si</b> imc &lt; 18.5 <b>Entonces</b>     <b>Escribir</b> "Peso Insuficiente";     ipe = ipe + 1; <b>FinSi</b> <b>Si</b> imc <math>\geq</math> 18.5 <b>y</b> imc <math>\leq</math> 24.9 <b>Entonces</b>     <b>Escribir</b> "Peso Normal";     pno = pno + 1; <b>FinSi</b> <b>Si</b> imc <math>\geq</math> 25 <b>y</b> imc <math>\leq</math> 29.9 <b>Entonces</b>     <b>Escribir</b> "SobrePeso";     sg = sg + 1; <b>FinSi</b> <b>Si</b> imc <math>\geq</math> 30 <b>y</b> imc <math>\leq</math> 34.9 <b>Entonces</b>     <b>Escribir</b> "Obesidad Grado I";     ob1 = ob1 + 1; <b>FinSi</b> <b>Si</b> imc <math>\geq</math> 35 <b>y</b> imc <math>\leq</math> 39.9 <b>Entonces</b>     <b>Escribir</b> "Obesidad Grado II";     ob2 = ob2 + 1; <b>FinSi</b> <b>Si</b> imc <math>\geq</math> 40 <b>Entonces</b>     <b>Escribir</b> "Obseidad Grado III";     ob3 = ob3 + 1; <b>FinSi</b>  <b>FinSubProceso</b>         </pre>

Acción	Resultado esperado
<p>El segundo subproceso determinará el total de pacientes en cada categoría y solo se mostrarán si son mayores a cero.</p>	<pre> SubProceso cantCliente(ipe,pno,sg,ob1,ob2,ob3) Si ipe &gt; 0 Entonces     Escribir "Cantidad de Clientes con Insuficiente Peso: ",ipe; FinSi Si pno &gt; 0 Entonces     Escribir "Cantidad de Clientes con Peso Normal: ",pno; FinSi Si sg &gt; 0 Entonces     Escribir "Cantidad de Clientes con SobrePeso Grado I: ",sg; FinSi Si ob1 &gt; 0 Entonces     Escribir "Cantidad de Clientes con Obesidad Grado I: ",ob1; FinSi Si ob2 &gt; 0 Entonces     Escribir "Cantidad de Clientes con Obesidad Grado II: ",ob2; FinSi Si ob3 &gt; 0 Entonces     Escribir "Cantidad de Clientes con Obesidad Grado III: ",ob3; FinSi FinSubProceso                     </pre>

### Práctica independiente

Se sugiere un trabajo en parejas o individual para escribir el tercer subproceso y describir los resultados de la situación de peso en kilogramos (kg) y de la altura en metros (m) de los pacientes de la clínica. Utilizando la fórmula para sacar el índice de masa corporal:

$$IMC = \frac{\text{Peso (kg)}}{\text{Altura}^2 \text{ (m)}}$$

Para poder determinar la categoría del paciente: PESO INSUFICIENTE, PESO NORMAL, SOBREPESO, OBESIDAD GRADO 1, OBESIDAD GRADO 2 y OBESIDAD GRADO 3.

Para lograrlo, se sugiere hacer una lectura comprensiva de lo realizado en la práctica guiada y rehacer si fuera necesario los subprocesos para comprender lo que falta del algoritmo.

Acción	Resultado esperado
<p>Por último, el tercer subproceso, el cual determinará en que categoría hay más clientes.</p>	<pre> <b>SubProceso</b> cantIMC(ipe,pno,sg,ob1,ob2,ob3)    <b>Si</b> ipe ≥ pno <b>y</b> ipe ≥ sg <b>y</b> ipe ≥ ob1 <b>y</b> ipe ≥ ob2 <b>y</b> ipe ≥ ob3 <b>Entonces</b>   ..... <b>Escribir</b> " Mas Clientes en Insuficiente Peso";   <b>FinSi</b>   <b>Si</b> pno ≥ ipe <b>y</b> pno ≥ sg <b>y</b> pno ≥ ob1 <b>y</b> pno ≥ ob2 <b>y</b> pno ≥ ob3 <b>Entonces</b>   ..... <b>Escribir</b> "Mas Clientes en Peso Normal";   <b>FinSi</b>   <b>Si</b> sg ≥ ipe <b>y</b> sg ≥ pno <b>y</b> sg ≥ ob1 <b>y</b> sg ≥ ob2 <b>y</b> sg ≥ ob3 <b>Entonces</b>   ..... <b>Escribir</b> "Mas Clientes en SobrePeso Grado I";   <b>FinSi</b>   <b>Si</b> ob1 ≥ ipe <b>y</b> ob1 ≥ pno <b>y</b> ob1 ≥ sg <b>y</b> ob1 ≥ ob2 <b>y</b> ob1 ≥ ob3 <b>Entonces</b>   ..... <b>Escribir</b> "Mas Clientes en Obesidad Grado I";   <b>FinSi</b>   <b>Si</b> ob2 ≥ ipe <b>y</b> ob2 ≥ pno <b>y</b> ob2 ≥ sg <b>y</b> ob2 ≥ ob1 <b>y</b> ob2 ≥ ob3 <b>Entonces</b>   ..... <b>Escribir</b> "Mas Clientes en Obesidad Grado II";   <b>FinSi</b>   <b>Si</b> ob3 ≥ ipe <b>y</b> ob3 ≥ pno <b>y</b> ob3 ≥ sg <b>y</b> ob3 ≥ ob1 <b>y</b> ob3 ≥ ob2 <b>Entonces</b>   ..... <b>Escribir</b> "Mas Clientes en Obesidad Grado III";   <b>FinSi</b>  <b>FinSubProceso</b>           </pre>
<p>Resultado</p>	<pre> Cantidad de clientes por categoria: ----- Cantidad de Clientes con Insuficiente Peso: 1 Cantidad de Clientes con Obesidad Grado I: 2 ----- La categoria con mas clientes es: ----- Mas Clientes en Obesidad Grado I ----- *** Ejecución Finalizada. ***           </pre>

## Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
<b>Códigos</b>	Elabora un código.	Elabora un código que cumple con lo establecido.	Elabora un código que cumple con lo establecido y también lo analiza y mejora en el proceso.
<b>Problemática</b>	Analiza el problema.	Analiza el problema y da una opinión al respecto.	Analiza el problema y da un argumento de cómo mejorarlo y optimizarlo.
<b>Condicionales</b>	Elabora las estructuras condicionales con errores.	Elabora las estructuras condicionales con algunos errores.	Elabora las estructuras condicionales sin errores.
<b>Repetidores o ciclos</b>	Elabora las estructuras cíclicas con errores.	Elabora las estructuras cíclicas con algunos errores.	Aplica las estructuras cíclicas sin errores.
<b>Subprocesos</b>	Elabora un subproceso.	Elabora de manera moderada un subproceso.	Elabora y aplica sin errores el uso de subproceso.

## Orientaciones al docente

**Para unificar conceptos disciplinares:** Con la nueva metodología, es posible estar de acuerdo que su función principal es dejar todo más organizado. Las personas al programar dividen las tareas en capas que van llamándolas cuando las necesite. Los subprocesos son eso, pero de una manera más simplificada, ya que este lenguaje no maneja el sistema donde se ven los programas de programación en otros ámbitos más allá de la educación. Debemos entender que los subprocesos son una complejidad agregada a lo que ya se ha visto anteriormente y que se puede implementar en todos los ejemplos en los cuales haga sentido utilizar esta función.

- Un subproceso o función no es más que un bloque de código con un determinado fin o propósito. Podemos definir más de uno y nos permiten **ahorrar líneas de código** si nuestros programas son muy extensos o hay codificación redundante.

**Actitudes:** Es recomendable el uso de computadores a la hora de codificar nuestro código. Más allá del papel, el computador se utiliza para verificar o evaluar de una manera más eficiente el código y también poder ejecutar y ver el funcionamiento de nuestro algoritmo. Para el desarrollo de los trabajos individuales, es necesario tener claro que esto es para que el estudiante tenga confianza de sus conocimientos y su forma de pensar. Por último, siempre es bueno ser críticos y honestos al momento de crear nuestros algoritmos, y siempre resolver de forma creativa los problemas que vayan apareciendo.

**Orientaciones para organizar e implementar la actividad:** Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Determinar un tiempo para así resolver cualquier duda del problema.
- Determinar el tiempo que cree necesario para haber terminado o al menos llevar algo de desarrollo de la actividad.
- Determinar el tiempo para retroalimentar a los estudiantes y guiar a quiénes presenten mayor dificultad.
- Favorecer que algún estudiante presente su resultado. Puede ser por voluntad propia o elegido por usted para poder dar una retroalimentación con refuerzo positivo.

## Actividad de desempeño 4

### Propósito

En esta actividad se busca que los estudiantes puedan evaluar y modelar algoritmos complejos mediante la metodología de los Arreglos. Su función primordial es guardar información y lo que más se asemeja a esto es el procesamiento de una hoja de cálculo tradicional.

### Objetivos de Aprendizaje

**OA2.** Organizar datos de una situación problemática, estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función.  
**(Evaluar)**

**OA3.** Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles.  
**(Modelar)**

### Conocimiento esencial

- Modelos entidad-relación y simulación de sistemas.
- Sistemas computacionales: dispositivos, *hardware* y *software*, y resolución de problemas.

### Tiempo estimado

3 horas pedagógicas.



## Desarrollo de la actividad

### Situación experiencial

Se sugiere presentar los arreglos como un nuevo tipo de dato para programar el promedio de notas, diferenciando entre el ingreso de un listado y la lectura de los datos numéricos de una tabla.



	A	B	C	D	E	F
1	Asignaturas	Nota 1	Nota 2	Nota 3	Nota 4	Nota 5
2	Arte	6,5	5,2	6,1	4,8	6,4
3	Matemática	5,1	6,9	5,8	4,5	6,2

Este arreglo tiene 3 filas, solo dos son numéricas.

Este arreglo tiene 6 columnas, solo 5 son numéricas.

Esta tabla es un nuevo tipo de dato. Se llama "arreglo" y se puede usar para programar, por ejemplo, el promedio de notas.

A partir de esta tabla, se sugiere realizar algunas preguntas que permitan reconocer el uso y lectura de una tabla:

- ¿Cómo se pueden leer los datos que están en la tabla?
- ¿Cómo podemos calcular el promedio?
- ¿Cuáles podrían ser la diferencia entre los algoritmos de listado de datos y de arreglos?

## Construcción de conocimiento

Para construir el conocimiento sobre la lectura de arreglos, se sugiere introducir los conceptos de filas y columnas a partir de la situación anterior.

	Columnas				
Filas	6,5	5,2	6,1	4,8	6,4
	5,1	6,9	5,8	4,5	6,2

En un pseudocódigo se anota primero el lugar de la fila y luego la posición de la columna  
 $m[\text{fila}][\text{columna}]$

La numeración de las filas y de las columnas empieza en 0.

$m[0][3] = 4,8$

$m[1][4] = 6,2$

$m[0][3] + m[1][4] = 4,8 + 6,2 = 11$

El cálculo del promedio de notas se puede hacer al menos de dos maneras. Una de ellas es calculando el promedio de cada asignatura y luego calculando el promedio de estas dos notas, o bien calculando inmediatamente un solo promedio. En estos casos, es necesario incluir el largo de cada fila o columna, por ejemplo, en el caso de una sola fila de notas:

6,5	5,2	6,1	4,8	6,4	5,1	6,9	5,8	4,5	6,2
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Para el arreglo unidimensional y la lectura de cada nota en pseudocódigo se utiliza  
 $d[\text{posiciones}]$

Al cual se le agrega el tipo de variable para poder hacer cálculos.

$d[0] = 6,5$

$d[3] = 4,8$

$d[0] + d[3] = 6,5 + 4,8 = 11,3$

Con el término *largo[d]* se calcula la cantidad de datos en la lista, que es necesario para calcular el promedio.

### Práctica guiada

Elaborar un programa en pseudocódigo que incluya las funciones “posiciones” y “largo”, y que permita calcular el promedio de las notas. Se sugiere estructurar el procedimiento de elaboración en 4 partes: en la primera se hace la definición del tipo de datos y las variables que serán consideradas. Para esto último, se sugiere considerar las partes que componen el promedio en suma en números racionales y la cantidad de sumandos en números enteros, que luego será el divisor. La segunda parte es el ingreso de datos utilizando la función posiciones, la tercera parte de la elaboración indica cómo se comienza con los datos según la posición, aquí es importante relevar el uso de índices. La última parte es la obtención del resultado y se sugiere evaluar estos resultados compartiendo el programa con otros e ingresando otros valores para ver cuál es el resultado.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar las definiciones de manera correcta	<p>Nombre programa: promedio de notas</p> <p><math>d[\text{posiciones}]</math>: números racionales</p> <p><math>sum</math>: números racionales</p> <p><math>i</math>: números enteros</p> <p><math>p</math>: números racionales</p> <p><math>\text{largo}[d]</math>: números enteros</p> <p><math>sum = 0.0</math></p>	¿Cómo podría empezar?
Elaborar el ingreso de datos	$d = \{6.5, 5.2, 6.1, 4.8, 6.4, 5.1, 6.9, 5.8, 4.5, 6.2\}$	¿Cómo hacer para guardar los arreglos?
Iniciar el programa	<p><b>Para <math>i = 0</math> hasta <math>i &lt; \text{largo}[d]</math></b></p> <p><math>i = i + 1</math></p> <p><math>sum = sum + d[i]</math></p> <p><b>Fin Para</b></p> <p><math>p = sum / \text{largo}[d]</math></p> <p>Escribir: “El promedio es <math>p</math>”</p>	¿Cómo puedo llamar a cada uno de los arreglos y ordenarlo?
Resultado esperado	$p = 5,75$	

## Práctica independiente

Se sugiere un trabajo de pares, donde ambos estudiantes resuelvan un ejercicio con arreglos. Para esto, se sugiere entregar una tabla para que el estudiante programe con un arreglo. En este caso, entregar información a la tabla de la base de datos.

CODIGO	NOMBRE	DIRECCIÓN	SEXO	SUELDO
COD01	ANA	CONCHALÍ	F	350000
COD02	LUIS	QUILICURA	M	375000
COD03	MARCELA	VITACURA	F	325000

El algoritmo debe ser capaz de agregar estos 3 datos y mostrarlos de la misma manera. Después de haber ingresado el primer dato, saldrá una ventana preguntando si desea ingresar otro registro. Al confirmarlo se repetirá la secuencia y se volverá a ingresar. Al rechazar la opción se mostrarán los registros que logró ingresar y finalizará el programa.

Considerar la lectura de la posición del dato según la fila y la columna para la programación utilizando un corchete cuadrado y anotando como par ordenado, donde la fila va en primer lugar y se comienza desde el cero. Se sugiere retomar lo trabajado en la construcción del conocimiento.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar las definiciones de manera correcta	<pre> <b>Algoritmo</b> agregar_registros  <b>Definir</b> continuar, datos <b>Como</b> <b>Caracter</b>; <b>Definir</b> f, c, fila <b>como</b> <b>numero</b>; <b>Dimension</b> datos[10,5];  continua ← "S"; f ← 0; c ← 0; fila ← 1; </pre>	¿Cómo podría empezar?
Elaborar el ingreso de datos	<pre> //Registrar Datos <b>Mientras</b> MAYUSCULAS(continua) == "S" <b>Hacer</b>   <b>Escribir Sin Saltar</b> "CODIGO   :"; <b>Leer</b> datos(fila,1);   <b>Escribir Sin Saltar</b> "NOMBRE   :"; <b>Leer</b> datos(fila,2);   <b>Escribir Sin Saltar</b> "DIRECCION :"; <b>Leer</b> datos(fila,3);   <b>Escribir Sin Saltar</b> "SEXO     :"; <b>Leer</b> datos(fila,4);   <b>Escribir Sin Saltar</b> "SUELDO   :"; <b>Leer</b> datos(fila,5);   fila ← fila + 1;   <b>Escribir Sin Saltar</b> "Desea Ingresar Otro Registro (S/N)? ";   <b>Leer</b> continua;   <b>Escribir</b> ""; <b>FinMientras</b> </pre>	¿Cómo hacer para que se guarde en los arreglos?

Acción	Resultado esperado	Preguntas de apoyo
<p>Listar los datos ingresados de manera ordenada</p>	<pre> //Listar Datos <b>Escribir</b> "REGISTROS: "; <b>Para</b> f ← 1 <b>hasta</b> fila-1 <b>con</b> <b>paso</b> 1 <b>hacer</b>   <b>Para</b> c ← 1 <b>hasta</b> 5 <b>con</b> <b>paso</b> 1 <b>hacer</b>     <b>Escribir Sin Saltar</b> datos[f,c], " ";   <b>FinPara</b> <b>Escribir</b> " "; <b>FinPara</b> <b>FinAlgoritmo</b>           </pre>	<p>¿Cómo puedo llamar a cada arreglo y ordenarlo?</p>
<p>Resultado esperado</p>	<pre> *** Ejecución Iniciada. *** CODIGO      :&gt; COD01 NOMBRE      :&gt; ANA DIRECCION   :&gt; CONCHALI SEXO        :&gt; F SUELDO      :&gt; 350000 Desea Ingresar Otro Registro (S/N)? &gt; S  CODIGO      :&gt; COD02 NOMBRE      :&gt; LUIS DIRECCION   :&gt; QUILICURA SEXO        :&gt; M SUELDO      :&gt; 375000 Desea Ingresar Otro Registro (S/N)? &gt; S  CODIGO      :&gt; COD03 NOMBRE      :&gt; MARCELA DIRECCION   :&gt; VITACURA SEXO        :&gt; F SUELDO      :&gt; 325000 Desea Ingresar Otro Registro (S/N)? &gt; N  REGISTROS: COD01   ANA   CONCHALI   F   350000   COD02   LUIS   QUILICURA   M   375000   COD03   MARCELA   VITACURA   F   325000   *** Ejecución Finalizada. ***           </pre>	

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Códigos	Elabora un código.	Elabora un código que cumple con lo establecido.	Elabora un código que cumple con lo establecido y lo analiza y mejora en el proceso.
Problemática	Analiza el problema.	Analiza el problema y da una opinión al respecto.	Analiza el problema y da un argumento de cómo mejorarlo y optimizarlo.
Arreglos	Elabora las estructuras de arreglos con errores.	Elabora las estructuras de arreglos con algunos errores.	Elabora las estructuras de arreglos sin errores.

### Orientaciones al docente

**Para unificar conceptos disciplinares:** Los arreglos son en general la parte más difícil de entender, pero si el conocimiento previo ha sido comprendido por los estudiantes, tendrán una buena base, pues, como es usual, siempre hay que practicarlo mediante ejercicios, ya que es muy importante a la hora de analizar algún problema que pueda tener una de estas formas de solucionarse. Hay que recordar que los arreglos son cajas que guardan información y cuando uno quiera ver esa información guardada, simplemente debe llamarla mediante el comando preciso para saber dicha información. En cuanto a las actividades, es necesario siempre estar pendiente de las dudas que puedan surgir y utilizar las herramientas aprendidas en los módulos anteriores.

- Un arreglo se define como una colección finita, homogénea y ordenada de elementos.
- Finita: porque todo arreglo tiene un límite, es decir, se debe determinar cuál será el número máximo de elementos que podrán formar parte del arreglo.
- Homogénea: porque todos los elementos del arreglo son del mismo tipo, por ejemplo: arreglo de enteros, de booleanos, pero nunca una combinación de distintos tipos.
- Ordenada: se puede determinar cuál es el primer elemento, el segundo elemento, el enésimo elemento.

El uso de la palabra sexo en este contexto es utilizado para identificar solo dos atributos, se sugiere ampliar esta categoría agregando otros atributos y utilizando la palabra género.

**Actitudes:** Es recomendable apoyar el desarrollo de estas actividades con el uso de un computador para que los estudiantes puedan ver sus algoritmos en tiempo real y ver el funcionamiento de estos. En cuanto al desarrollo de la actividad individual, es necesario siempre responder las consultas y guiar a los estudiantes, pero sobre todo favorecer el trabajo en equipo y que los estudiantes se ayuden mutuamente a resolver los problemas que tengan por delante.

**Orientaciones para organizar e implementar actividades:** Se sugieren las siguientes indicaciones para guiar la implementación de las actividades:

- Elaborar actividades donde se utilicen los arreglos.
- Determinar el tiempo necesario para elaborar el análisis y desarrollo del algoritmo.
- Concluir mostrando algunos de los algoritmos de los estudiantes para que sea revisado por todos los estudiantes (esto con el fin de mostrar la forma en la que lo hicieron y ver si algún otro estudiante lo habría hecho de otra manera).

## Módulo obligatorio 3

### Visión panorámica

#### Gran idea

El comportamiento de un sistema computacional (dispositivos, *hardware* y *software*) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema.

#### Objetivos de aprendizaje

- OA4.** Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. **(Generalizar y Transferir)**
- OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y Transferir)**
- OA6.** Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

#### Conocimientos esenciales

- Redes e Internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.
- Algoritmos y programación: Lenguajes, métodos y entornos de programación.

#### Tiempo estimado

6 semanas (12 horas)

## Propósito Módulo obligatorio 3

En el módulo 3 de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media, se espera que los estudiantes puedan aplicar en diferentes escenarios los conocimientos asociados al comportamiento de los sistemas computacionales y distinguir el rol que les cabe como usuarios al automatizar procesos lógicos para desempeñar o ejecutar tareas que resuelvan problemas de seguridad, acceso y manejo de datos en un ambiente digital. La adquisición de conocimientos asociados al manejo de sistemas de información cobra fuerza en este módulo, al ser los estudiantes quienes comienzan a transferir sus aprendizajes a estos ámbitos de las ciencias de la computación.

Sobre este escenario, el módulo pone énfasis en desarrollar las habilidades de generalización y transferencia a través del pensamiento algorítmico y la programación propiamente tal, ya que al tener conocimientos sobre tablas y modelado de datos, desde la descomposición, el modelado, la abstracción y la evaluación, los estudiantes podrán trabajar con información en distintos lenguajes y manejarlos para poder utilizar de forma eficiente distintos datos y fuentes, los cuales están relacionados con problemas de seguridad, acceso y control. Asimismo, los estudiantes podrán comprender que, a través de las actividades de desempeño del módulo, los sistemas computacionales pueden estar vinculados a redes e internet, distinguiendo que el uso de sitios web, aplicaciones móviles y objetos controlados digitalmente siempre tienen a la informática y sus diferentes clasificaciones para determinar o evaluar su funcionamiento.

Los Objetivos de Aprendizaje del módulo desarrollan las actitudes del siglo XXI del ámbito de las Maneras de Trabajar y Herramientas para trabajar promoviendo el aprovechamiento de las herramientas para resolver problemas o diseñar de forma colaborativa soluciones para el logro de metas comunes. Asimismo, al aprovechar las herramientas disponibles, debe enfocarse en que las mismas no son siempre limitantes para poder crear soluciones a problemas cotidianos, pues ofrecen de distintas formas oportunidades de aprendizaje. Finalmente, las actividades de desempeño propuestas esperan ser una oportunidad para comprender que las interacciones de los estudiantes con las tecnologías tienen efectos en ellos mismos y la sociedad, pues las mismas son sujetas de valoración y nunca son neutras.



## Ruta de aprendizaje del Módulo 3

### ¿Cómo distinguir la tipología de sistemas y redes para solucionar problemas de seguridad, acceso y control de datos?

#### Actividad de desempeño 1:

Resolver problemas computacionales relacionados con la funcionalidad de un objeto relevando la función de programación “hasta que”.

#### Actividad de desempeño 2:

Comparar programas según las funciones de programación y lazos utilizados en su elaboración.

**Actividad de desempeño 3:** Utilizar las oportunidades que pueden ofrecer las bases de datos en ámbitos de desempeño profesional o cotidiano.



#### Actividad de desempeño

**4:** Aplicar soluciones computacionales de forma flexible a diferentes problemas, transfiriendo los conocimientos asociados al uso de base de datos en estructuras informacionales más complejas.

## Actividad de desempeño 1

### Propósito

Esta actividad desafía a los estudiantes a volver a los orígenes de la formulación de algoritmos, abordándolos desde un enfoque más técnico, secuencial y abstracto, favoreciendo el desarrollo del pensamiento computacional y definiendo los pasos que se desencadenan para la consecución de una acción a partir de una instrucción. Además de integrar y movilizar los aprendizajes adquiridos previamente, se busca que los estudiantes puedan resolver situaciones complejas a nivel computacional de forma progresiva, en cuanto al nivel de transferencia y generalización.

### Objetivos de Aprendizaje

**OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y Transferir)**

**OA6.** Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

### Conocimiento esencial

- Algoritmos y programación: Lenguajes, métodos y entornos de programación.

### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

Se sugiere que el docente explique la importancia de la tecnología y lo inherente que es al desarrollo del ser humano, poniendo énfasis en la velocidad con la cual ésta ha avanzado durante los últimos años. Es posible considerar como contexto que, por ejemplo, el primer mensaje de texto enviado en el mundo fue el 3 de diciembre de 1992 desde un computador a un teléfono móvil y que hoy en día una gran cantidad de personas en el mundo utiliza WhatsApp, una aplicación de mensajería instantánea, como principal medio de comunicación.



Algunas de las preguntas que podrían promover procesos de pensamiento asociados a la argumentación y reflexión sobre avances tecnológicos asociados al uso de internet y redes a través del tiempo, podrían ser:

- ¿Qué piensan sobre los avances de Internet durante los últimos 10 años?
- ¿Cuáles son las evidencias que ustedes tienen para indicar que la tecnología ha avanzado?
- ¿Cómo serían las comunicaciones hoy en día si no se hubiese inventado internet?
- ¿De qué manera imaginas el avance de la tecnología durante los próximos 10 años?

### Construcción de conocimiento

Para seguir profundizando en el desarrollo del pensamiento algorítmico, se sugiere recuperar los fundamentos básicos aprendidos y cómo estos ayudan a resolver cualquier problema que se presente a un usuario o creador de tecnología. Para esta actividad, el docente puede dar la instrucción de crear un algoritmo en base a necesidades cotidianas o la funcionalidad de un objeto, por ejemplo, lavarse los dientes, ir a trabajar por la mañana, funcionamiento de un reloj, entre otras.

Para el caso de un reloj, se creará un cronómetro que funcione con “hora: minuto: segundo”, la idea principal es que se ingrese qué hora, minuto y segundo deba contar el cronómetro hasta llegar a 0. Después de eso, se le preguntará al usuario si desea repetir la operación.

Después de haber hecho el pseudocódigo, será necesario analizarlo y evaluarlo de acuerdo con su funcionamiento.

Acción	Resultado esperado	Preguntas de apoyo
<p>Elaborar el inicio del pseudocódigo y los datos de cuánto durará el cronómetro</p>	<pre> <b>Algoritmo</b> Cronometro  <b>Definir</b> H, M, S, CH, CM, Cs <b>como numero</b>; <b>Definir</b> continuar <b>como caracter</b>; continuar ← "s";  <b>Mientras</b> continuar ="s" <b>Hacer</b>   <b>Repetir</b>     <b>Escribir</b> "Ingrese la hora : ";     <b>Leer</b> H;   <b>Hasta Que</b> H ≤ 24;   <b>Repetir</b>     <b>Escribir</b> "Ingrese el minuto : ";     <b>Leer</b> M;   <b>Hasta Que</b> M ≤ 60;   <b>Repetir</b>     <b>Escribir</b> "Ingrese el segundo : ";     <b>Leer</b> S;   <b>Hasta Que</b> S ≤ 60; </pre>	<p>¿Qué algoritmo podría hacer?</p>
<p>Realizar el algoritmo complejo para completar la función de cronómetro</p>	<pre> <b>Escribir</b> "TIEMPO INGRESADO : ", H, ":", M, ":", S; <b>Para</b> CH ← 0 <b>hasta</b> H <b>con paso 1 hacer</b>   <b>para</b> CM ← 0 <b>hasta</b> M <b>con paso 1 hacer</b>     <b>para</b> CS ← 0 <b>hasta</b> S <b>con paso 1 hacer</b>       <b>Escribir</b> (H-CH), ":", M, ":", (S-CS);       <b>Esperar</b> 1 Segundo;     <b>FinPara</b>     S ← 60;     M ← M -1;   <b>FinPara</b>   M ← 60;   S ← 60; <b>FinPara</b>  <b>Escribir</b> "FIN DE CRONÓMETRO"; <b>Escribir</b> "Volvera a Reiniciar la cuenta? (s/n)"; <b>Leer</b> continuar;  <b>FinMientras</b> <b>FinAlgoritmo</b> </pre>	<p>¿El algoritmo debe ser complejo?</p>

## Práctica guiada

Para guiar el proceso de Evaluar, se sugiere que el docente retome el ejercicio “PC-Masters” del módulo anterior, pero esta vez utilizar los arreglos para guardar los artículos que se introducirán, como máximo 10, y al finalizar se hará una factura donde además se sumará el IVA total de los productos comprados. Se trabaja con el IVA de un 18%.

Acción	Resultado esperado
<p>Primero se empezará creando las dimensiones para guardar las diferentes categorías que guardará el producto, esto debido a que solo guardan una clase de datos (carácter o numérico, pero no ambos).</p>	<pre> <b>Algoritmo</b> PCMastersV2   <b>Dimension</b> articulo[10,1], cantidad[10,1], precio[10,1], total[10,1]   Empresa = "PCMasters"   continuar ← "S";   fila ← 1;   //Inicio   <b>Escribir</b> "INGRESE DATOS DEL CLIENTE"   <b>Escribir</b> "======"   <b>Escribir Sin Saltar</b> "Nombre   : "   <b>Leer</b> Nombre   <b>Escribir Sin Saltar</b> "RUT     : "   <b>Leer</b> Rut   <b>Escribir Sin Saltar</b> "Direccion : "   <b>Leer</b> Direccion           </pre>
<p>Luego, crear un <b>mientras</b> se mantenga el carácter “s”, para poder repetir en cada momento la acción y así agregar los nuevos artículos.</p>	<pre> //Desarrollo <b>Escribir</b> "INGRESE ARTICULOS" <b>Escribir</b> "======" <b>Mientras</b> MAYUSCULAS(continuar) == "S" <b>Hacer</b>   <b>Escribir Sin Saltar</b> "Articulo Nro " fila " : "; <b>Leer</b> articulo(fila,1);   <b>Escribir Sin Saltar</b> "Cantidad      : "; <b>Leer</b> cantidad(fila,1);   <b>Escribir Sin Saltar</b> "Pre. Unitario \$ : "; <b>Leer</b> precio(fila,1);   fila ← fila + 1;   <b>Escribir Sin Saltar</b> "OTRO ARTICULO (S/N)?"; <b>Leer</b> continuar;   <b>Escribir</b> ""; <b>FinMientras</b>           </pre>

Acción	Resultado esperado
<p>Crear el menú de la factura indicando todos los datos del comprador ingresados anteriormente y utilizar la función para y hasta con las diferentes dimensiones; hacer el respectivo cálculo del precio y guardar el total como una suma.</p>	<pre> <b>Escribir</b> "FACTURA Nro 000-1" <b>Escribir</b> "===== " <b>Escribir</b> "Empresa : " Empresa <b>Escribir</b> "Cliente : " Nombre " RUT : " Rut <b>Escribir</b> "Direccion : " Direccion <b>Escribir</b> "===== " <b>Escribir</b> "CANTIDAD DESCRIPCION PRE. UNIT. PRE. TOTAL" <b>Para</b> f ← 1 <b>hasta</b> fila-1 <b>con</b> paso 1 <b>hacer</b>   <b>Para</b> c ← 1 <b>hasta</b> 1 <b>con</b> paso 1 <b>hacer</b>     <b>Escribir Sin Saltar</b> cantidad[f,c], " ";     <b>Escribir Sin Saltar</b> " " articulo[f,c], " ";     <b>Escribir Sin Saltar</b> " \$" precio[f,c], " ";     total[f,c] = cantidad[f,c] * precio[f,c]     TotalT = TotalT + total[f,c]     <b>Escribir Sin Saltar</b> " \$" total[f,c], " ";   <b>FinPara</b> <b>Escribir</b> " "; <b>FinPara</b> </pre>
<p>Al finalizar, mostrar el costo total y el IVA, además de la suma de estos dos para dar el fin de la factura con el total a pagar.</p>	<pre> <b>Escribir</b> "===== " IVA = TotalT * 0.18 <b>Escribir</b> " \$" TotalT <b>Escribir</b> " IMPUESTO IVA 18% \$" IVA TotalTT = TotalT + IVA <b>Escribir</b> " ----- " <b>Escribir</b> " TOTAL A PAGAR \$" TotalTT  <b>FinAlgoritmo</b> </pre>

Acción	Resultado esperado
Resultado de la operación.	<pre> INGRESE DATOS DEL CLIENTE ===== Nombre   :&gt; Harry Potter RUT      :&gt; 168457899 Direccion :&gt; Av Siempre Viva 1234 INGRESE ARTICULOS ===== Articulo Nro 1  :&gt; Intel core i5 Cantidad       :&gt; 2 Pre. Unitario \$ :&gt; 189990 OTRO ARTICULO (S/N)?&gt; s  Articulo Nro 2  :&gt; RTX 3080Ti Cantidad       :&gt; 3 Pre. Unitario \$ :&gt; 899990 OTRO ARTICULO (S/N)?&gt; s  Articulo Nro 3  :&gt; ROG Helios Cantidad       :&gt; 1 Pre. Unitario \$ :&gt; 299990 OTRO ARTICULO (S/N)?&gt; n  FACTURA Nro 000-1 ===== Empresa   : PCMasters Cliente   : Harry Potter                RUT : 168457899 Direccion : Av Siempre Viva 1234 ===== CANTIDAD  DESCRIPCION      PRE. UNIT.      PRE. TOTAL 2         Intel core i5        \$.189990        \$.379980 3         RTX 3080Ti          \$.899990        \$.2699970 1         ROG Helios         \$.299990        \$.299990 =====  \$.3379940 IMPUESTO IVA 18% \$.608389.2 ----- TOTAL A PAGAR \$.3988329.2000000002 </pre>

### Práctica independiente

Se recomienda un trabajo entre pares en donde cada estudiante proponga un cambio en la situación de la factura con el IVA y realizar la programación en pseudocódigo utilizando el mismo esquema anterior. Algunos de los cambios que se sugieren son: variar la cantidad de artículos, el porcentaje de IVA, no incluir la dirección del cliente o agregar un nuevo dato del cliente, o bien calcular primero el IVA de cada artículo y luego el IVA total.

Variación	Marca inicial en el algoritmo
Cantidad de artículos	<p><b>Algoritmo</b> PCMastersV2</p> <pre> Dimension articulo[10,], cantidad[10,], precio[10,], total[10,] Empresa = "PCMasters" continuar ← "S"; fila ← 1; </pre>
Sin dirección	<pre> Escribir "INGRESE DATOS DEL CLIENTE" Escribir "===== " Escribir Sin Saltar "Nombre  : " Leer Nombre Escribir Sin Saltar "RUT      : " Leer Rut Escribir Sin Saltar "Direccion : " Leer Direccion </pre>
Cambio en el porcentaje	<pre> Escribir "===== " IVA = TotalT * 0.18 Escribir " Escribir " Escribir " TotalTT = TotalT + IVA Escribir " Escribir " </pre> <div style="text-align: right;"> <pre> \$. " TotalT IMPUERTO IVA 18% \$ " IVA ----- " TOTAL A PAGAR \$. " TotalTT </pre> </div>

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Pensamiento algorítmico	Formula una problemática algorítmica simple.	Formula un problema algorítmico medianamente complejo.	Formula un problema algorítmico complejo.
Seudocódigo	Elabora un pseudocódigo con dificultad para ejecutarse.	Elabora un pseudocódigo que logra ejecutarse de forma parcial.	Elabora un pseudocódigo que logra ejecutarse de forma completa.



## Orientaciones al docente

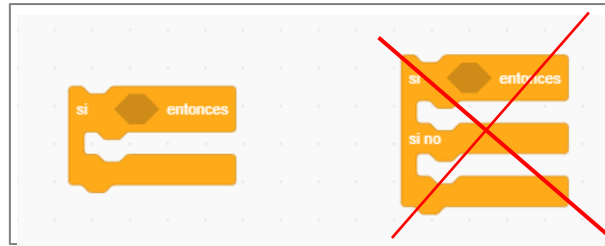
**Para unificar conceptos disciplinares:** En cuanto a pensar de forma algorítmica, se refiere a la forma de poder ver las acciones cotidianas y convertirlas en algoritmos y así dar una serie de instrucciones para interpretarlo de esta manera. Además, es necesario tener claros todos los conceptos que conlleva crear un algoritmo para poder crear los propios con el fin de resolver problemas y así comprender mejor lo que conlleva pensar de esta forma.

Pensar de forma algorítmica es la capacidad para expresar soluciones de forma tal que se componga de una serie de pasos finitos, no ambiguos y ordenados que permitan que una persona o un procesador de información pueda llevarlo adelante.

Se sugiere, en caso de ser posible, incluir algunos elementos de la programación en bloques, para ver alcances y limitantes en la creación de un cronómetro de cuenta regresiva. En este ejemplo, se ha utilizado el programa *Scratch*.

The image shows two instances of Scratch code for a countdown timer. The top instance shows the initial state where the timer is set to 60. The bottom instance shows the timer at 47 after 13 seconds. Annotations on the right side of the image explain the code components: the 'cronómetro' variable is an integer, the loop is a 'Mientras' loop, and the timer value is shown decreasing from 60 to 47 after 13 seconds.

La función lazo “*mientras*” da la condición de “*correr el programa*” que termina automáticamente cuando la condición no se cumple más. En este caso el lazo no necesita una condición de “*si no*”.

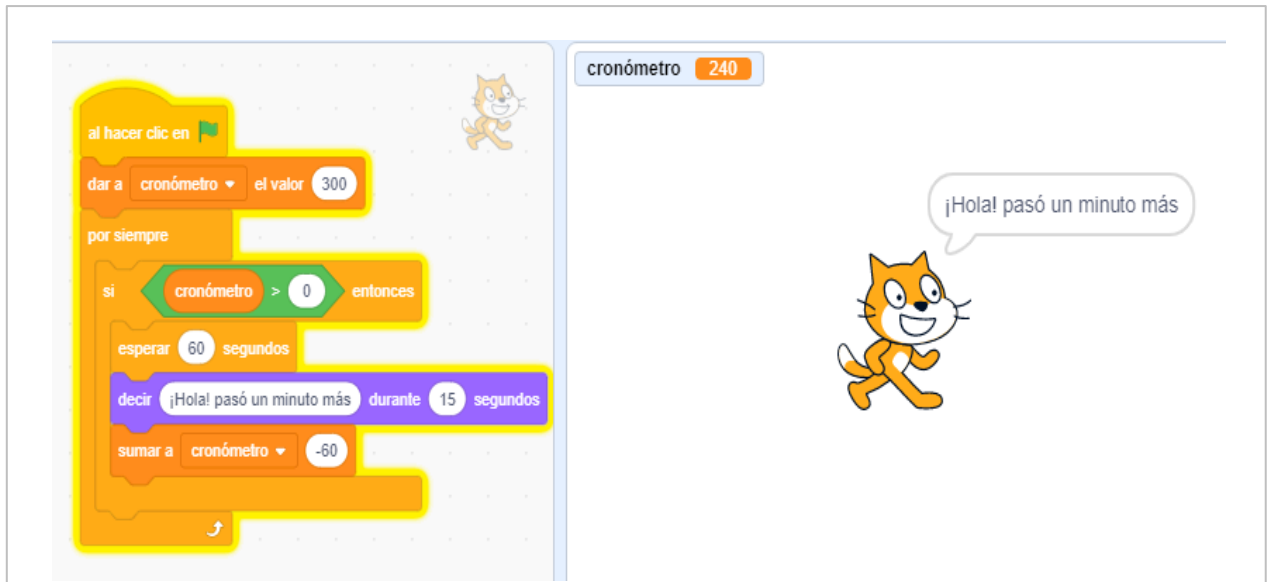


Se sugiere incluir algunas preguntas que permitan agudizar la lectura de programas en pseudocódigo y en bloques, por ejemplo, detectando que el cronómetro indica a partir de 60 segundos y cada 5 segundos el tiempo restante.

¿En qué modo está funcionando el cronómetro que describe el programa en “*scratch*”?



También se pueden incluir nuevas instrucciones tales como elaborar un programa en *scratch* que muestre un cronómetro que cuenta hacia atrás a partir de 300 segundos mostrando cada 60 segundos la frase “¡Hola! Pasó un minuto más”. Y solicitando que este mensaje deba aparecer por 15 segundos.



**Actitudes:** Para apoyar el desarrollo de las actividades, cuando se requiera de un nivel complejo de codificación, es recomendable el uso de computadores para facilitar el desarrollo de estos ejercicios. Por otra parte, se puede utilizar una hoja de papel o un documento Word para los problemas más simples en cuanto a pensamiento algorítmico.

**Orientaciones para organizar e implementar el trabajo colaborativo:** se sugieren las siguientes indicaciones para guiar la implementación del trabajo entre pares:

- Dejar que los alumnos elijan sus pares o que el docente elija las parejas y explicarles el concepto del desafío (que el alumno 1 haga su propio problema para que el alumno 2 lo resuelva y viceversa).
- Los alumnos elaborarán sus desafíos y se los explicarán para que puedan resolverlo de manera algorítmica.
- Determinar el tiempo necesario para elaborar un algoritmo complejo.
- Organizar el espacio para el trabajo en pares.
- Determinar el tiempo para que ambos estudiantes revisen el algoritmo del otro. Dar una retroalimentación apoyados por una rúbrica previamente conocida.

## Actividad de desempeño 2

### Propósito

Para esta actividad se llevará a cabo la ampliación del conocimiento en base al funcionamiento de los algoritmos. Esto conlleva a desafiar a los estudiantes a tomar conciencia del nivel de dificultad que llegan a tener los problemas y la magnitud de las soluciones, lo cual implica pensar y poner en uso habilidades y conocimientos posibles de seguir transfiriendo a otras realidades. En la medida que manejan más herramientas para plantear algoritmos, se suman más procedimientos y acciones que aumentan la eficiencia al pensar en soluciones frente a un problema.

### Objetivos de Aprendizaje

**OA4.** Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. **(Generalizar y Transferir)**

**OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y Transferir)**

### Conocimiento esencial

Algoritmos y programación: Lenguajes, métodos y entornos de programación.

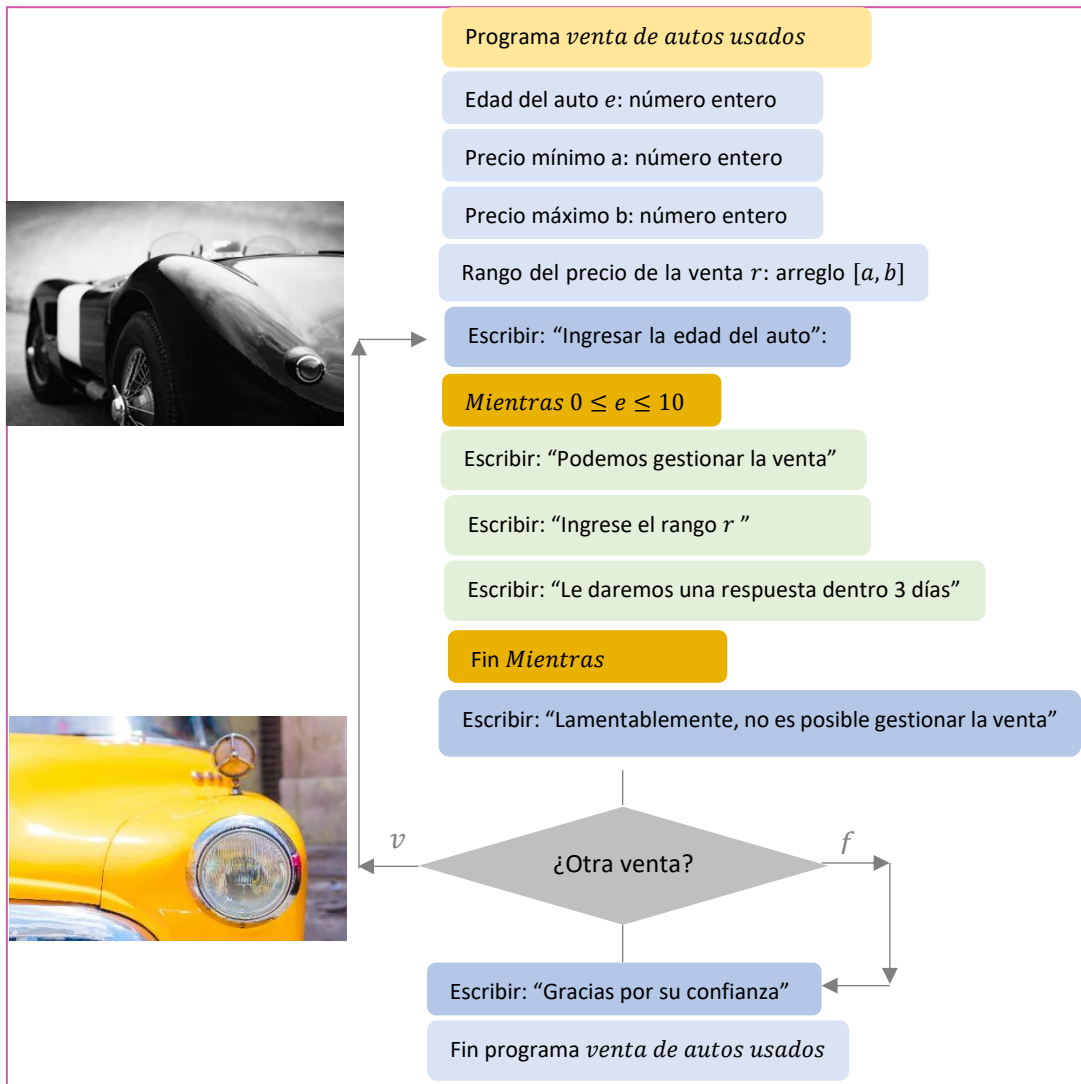
### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

El docente presenta una situación de venta de autos usados y la creación de un programa en pseudocódigo que puede servir como plataforma virtual. La idea es que los usuarios de la plataforma pueden ingresar la edad del auto y el programa acepta gestionar la venta, si no se supera la edad de 10 años del auto. Después, los usuarios pueden ingresar los datos del auto en venta. Se puede ingresar más de un auto en venta. Si el auto supera la edad de 10 años, se da el mensaje “Lamentablemente no podemos gestionar la venta de su auto”.



Algunas de las preguntas que podrían promover la conversación en torno a los lazos y funciones de programación son:

- ¿Cómo describirías este programa?
- ¿Cuál es el problema al cuál se quiere responder?
- ¿Cómo funciona este programa?

### Construcción de conocimiento

Para construir el conocimiento, se sugiere comparar dos programas diferentes, pero con el mismo objetivo, en este caso: elaborar una calculadora. En ambos programas, se necesitan diferenciar los lazos y las funciones de programación que se han utilizado en los programas.

Acción	Resultado esperado	Preguntas de apoyo
Elaborar las definiciones de manera correcta	<pre> <b>Algoritmo</b> Calculadora <b>Definir</b> opc, valor1, valor2 <b>Como Entero</b>; <b>Definir</b> total <b>como real</b>; <b>Escribir</b> "Calculadora"; <b>Escribir</b> "1: Sumar"; <b>Escribir</b> "2: Restar"; <b>Escribir</b> "3: Multiplicar"; <b>Escribir</b> "4: Dividir"; <b>Escribir</b> "Ingrese opcion:"; <b>Leer</b> opc;           </pre>	<p>¿Cuál es el nombre del programa?</p> <p>¿De qué tipo son las variables?</p> <p>¿Cómo lee el programa estas palabras?</p>
Reconocer los subprocesos, lazos y funciones	<pre> <b>Si</b> opc ≥ 1 <b>y</b> opc ≤ 4 <b>Entonces</b>   <b>Escribir</b> "Ingrese valor 1: ";   <b>Leer</b> valor1;    <b>Escribir</b> "Ingrese valor 2: ";   <b>Leer</b> valor2;    <b>Segun</b> opc     1:       total = valor1 + valor2;       <b>Escribir</b> "La suma de " valor1 " + " valor2 " = " total;     2:       total = valor1 - valor2;       <b>Escribir</b> "La resta de " valor1 " - " valor2 " = " total;     3:       total = valor1 * valor2;       <b>Escribir</b> "La multiplicacion de " valor1 " * " valor2 " = " total;     4:       total = valor1 / valor2;       <b>Escribir</b> "La division de " valor1 " / " valor2 " = " total;   <b>FinSegun</b>   <b>SiNo</b>     <b>Escribir</b> "Opcion invalida";   <b>FinSi</b>  <b>FinAlgoritmo</b>           </pre>	<p>¿Dónde se utiliza la función "si...entonces"?</p> <p>¿Dónde comienza la función "según"?</p> <p>¿Dónde termina la función "si... entonces"?</p> <p>¿Está incluida la función "para"?</p> <p>¿Qué tipos de subprocesos están incluidos?</p>

Para comparar, se sugiere utilizar las mismas preguntas de apoyo y anotar las discrepancias, para luego ver las diferencias en el funcionamiento del programa.

Acción	Resultado esperado	Preguntas de apoyo
<p>Elaborar las definiciones de manera correcta</p>	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <p><b>Algoritmo</b> CalculadoraV2</p> <p><b>Dimension</b> Num[1,10], Signos[1,10]</p> </div>	<p>¿Cuál es el nombre del programa?</p> <p>¿De qué tipo son las variables?</p> <p>¿Cómo lee el programa estas palabras?</p>
<p>Reconocer los subprocesos, lazos y funciones</p>	<pre> Repetir   Limpiar Pantalla   TotalT = 0;   Escribir "Calculadora";   Escribir Sin Saltar "Ingrese el 1 numero: "; Leer Num[1,1];   TotalT = TotalT + Num[1,1];   Escribir Sin Saltar "Cuantas operaciones planea hacer? (Maximo 10)"; Leer f;   Escribir "";   Para c ← 2 hasta f con paso 1 hacer     Escribir "1: Sumar - 2: Restar - 3: Multiplicar - 4: Dividir";     Escribir Sin Saltar "Ingrese opcion:"; Leer opc;     Segun opc       1: Signos[1,c] = " + "       2: Signos[1,c] = " - "       3: Signos[1,c] = " * "       4: Signos[1,c] = " / "     FinSegun     Escribir Sin Saltar "Ingrese el " c " numero: "; Leer Num[1,c];     Segun opc       1: TotalT = TotalT + Num[1,c];       2: TotalT = TotalT - Num[1,c];       3: TotalT = TotalT * Num[1,c];       4: TotalT = TotalT / Num[1,c];     FinSegun   FinPara   Escribir Sin Saltar " = " TotalT; Escribir "";   Escribir Sin Saltar "OTRA OPERACION (S/N)?"; Leer aux;   Escribir "Precione una tecla"   Esperar Tecla Hasta Que Mayusculas(aux) == "N" FinAlgoritmo           </pre>	<p>¿Dónde se utiliza la función “si... entonces”?</p> <p>¿Dónde comienza la función “según”?</p> <p>¿Dónde termina la función “si... entonces”?</p> <p>¿Está incluida la función “para”?</p> <p>¿Qué tipos de subprocesos están incluidos?</p>

### Práctica guiada

Se sugiere proponer a los estudiantes volver a recrear el algoritmo para calcular el promedio final. Para esto deben tener claro que se preguntará el nombre del alumno y sus cuatro respectivas notas (tres notas parciales y una nota final). Después de haber ingresado las notas, se le preguntará si desea agregar otro alumno. Al finalizar, se pondrá el nombre con el promedio final del alumno.

$$\text{Promedio Nota} = \frac{\text{suma de todas las notas}}{\text{cantidad de notas}}$$

$$\text{Promedio final} = (\text{Promedio Nota} \cdot 0,6) + (\text{Final} \cdot 0,4)$$

Acción	Resultado esperado
<p>Elaborar el ingreso de datos mediante el guardado de información en los arreglos.</p>	<pre> <b>Algoritmo</b> CalcularPromedioFinalV2 <b>Dimension</b> Alumno[10,1], Nota[10,3], promedio[10,1], NT[10,1] continuar ← "S"; fila ← 1; <b>Mientras</b> MAYUSCULAS(continuar) == "S" <b>Hacer</b>     <b>Escribir Sin Saltar</b> "Nombre Alumno " fila " : ";      <b>Leer</b> Alumno(fila,1)     <b>Para</b> c ← 1 <b>hasta</b> 3 <b>con paso</b> 1 <b>hacer</b>         <b>Escribir Sin Saltar</b> "Nota " c " : ";      <b>Leer</b> Nota(fila,c);     <b>FinPara</b>     <b>Escribir Sin Saltar</b> "Nota Examen Final: ";      <b>Leer</b> NT(fila,1);     fila ← fila + 1;     <b>Escribir Sin Saltar</b> "OTRO ALUMNO (S/N)?" ;      <b>Leer</b> continuar;     <b>Escribir</b> " "; <b>FinMientras</b>                 </pre>
<p>Utilizar los arreglos guardados para hacer los respectivos cálculos y proporcionar la respectiva visualización del nombre y promedio final.</p>	<pre> <b>Escribir</b> "NOMBRE          PROMEDIO"; <b>Para</b> f ← 1 <b>hasta</b> fila-1 <b>con paso</b> 1 <b>hacer</b>     <b>Para</b> c ← 1 <b>hasta</b> 3 <b>con paso</b> 1 <b>hacer</b>         promedio[f,1] = promedio[f,1] + Nota[f,c];     <b>FinPara</b>     promedio[f,1] = ((promedio[f,1] / 3) * 0.6 + (NT[f,1] * 0.4))      <b>Escribir Sin Saltar</b> Alumno[f,1] "          ";     <b>Escribir Sin Saltar</b> promedio[f,1],;     <b>Escribir</b> " "; <b>FinPara</b> <b>FinAlgoritmo</b>                 </pre>



Acción	Resultado esperado							
Resultado esperado tras la ejecución del algoritmo.	<pre>Nombre Alumno 1:&gt; Wolverine Nota 1:&gt; 20 Nota 2:&gt; 55 Nota 3:&gt; 40 Nota Examen Final: &gt; 45 OTRO ALUMNO (S/N)?&gt; s</pre>	<pre>Nombre Alumno 2:&gt; Peter Parker Nota 1:&gt; 60 Nota 2:&gt; 65 Nota 3:&gt; 60 Nota Examen Final: &gt; 70 OTRO ALUMNO (S/N)?&gt; n</pre>						
	<table border="1"> <thead> <tr> <th>NOMBRE</th> <th>PROMEDIO</th> </tr> </thead> <tbody> <tr> <td>Wolverine</td> <td>41</td> </tr> <tr> <td>Peter Parker</td> <td>65</td> </tr> </tbody> </table>		NOMBRE	PROMEDIO	Wolverine	41	Peter Parker	65
NOMBRE	PROMEDIO							
Wolverine	41							
Peter Parker	65							

Se sugiere que el docente favorezca que los estudiantes sitúen el foco en las mejoras respecto los aprendizajes para elaborar un pseudocódigo utilizando las bases del algoritmo, destacando que esto les permite elaborar soluciones mucho más complejas y así llevar a cabo soluciones que abarquen un mayor número de posibilidades.

Luego, podría indicar a los estudiantes que en otras circunstancias las formas de formular un algoritmo pueden ser ilimitadas. Algunas se parecerán en mayor o menor medida entre otras, pero siempre se intentará resolver el mismo problema, por eso es necesario que los estudiantes puedan tener claro que lo importante es siempre resolver el problema.

### Práctica independiente

Se sugiere un trabajo individual de evaluación y la transferencia de algoritmos. Para esto, se pueden considerar dos programas en pseudocódigo y código: el pseudocódigo utilizando el *software* ya antes trabajado PSeint u otro formato y como código el lenguaje de programación Python, en los cuales se hace una agregación de datos a una lista o arreglo y donde se pondrá información como nombre, apellido y dirección.

Acción	Resultado esperado
<p>Ver que es un ejercicio donde se guarda información en un arreglo y después se muestra la información ingresada.</p>	<pre> <b>Algoritmo</b> Listado   Definir datos como Caracter;   Definir f, c como Entero;    Dimension datos[3,3];    Para f ← 1 hasta 2 con paso 1 hacer     Escribir Sin Saltar "NOMBRE : "; Leer datos[f,1];     Escribir Sin Saltar "APELLIDO : "; Leer datos[f,2];     Escribir Sin Saltar "DIRECCIÓN : "; Leer datos[f,3];     Escribir "";   FinPara    Escribir "";   Escribir "NOMBRE   APELLIDO   DIRECCION"   Para f ← 1 hasta 2 con paso 1 hacer     Escribir datos[f,1],"   ",datos[f,2],"   ",datos[f,3];   FinPara  <b>FinAlgoritmo</b>           </pre>
<p>Esto es una versión más formal de lo que está arriba, entonces apreciar aquí que, a pesar de tener diferencias, tampoco es que se vean muy diferentes, ya que hacen exactamente lo mismo.</p>	<pre> if __name__ == '__main__':     datos = [[str() for ind0 in range(3)] for ind1 in range(3)]      for f in range(3):         print("NOMBRE : ")         datos[f][0] = input()         print("APELLIDO : ")         datos[f][1] = input()         print("DIRECCIÓN : ")         datos[f][2] = input()         print("")      print("")     for f in range(3):         print(datos[f][0], " ", datos[f][1], " ", datos[f][2])           </pre>

## Evaluación formativa

Para verificar aprendizajes, se sugiere utilizar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Seudocódigo	Elabora unseudocódigo.	Elaboro un código que cumplía con lo establecido.	Elaboro un código que cumplía con lo establecido e incorpora mejoras.
Solución	Mejora elseudocódigo.	Mejora la solución incorporando una opción no descrita en el problema.	Mejora la solución incorporando opciones que mejoran lo establecido en el problema y la solución.
Análisis al problema	Analiza el problema sin mejoras.	Analiza el problema y entrega parcialmente mejoras.	Analiza el problema y entrega mejoras para optimizarlo.

## Orientaciones al docente

**Para unificar conceptos disciplinares:** Se plantea que en este módulo se propongan desafíos o ejercicios más complejos y motivar a los estudiantes a pensar más allá del problema para poder resolver. Además, es necesario realizar una revisión de lo que es un lenguaje de programación computacional para trabajar, de preferencia, con Python porque es un lenguaje sencillo y muy recomendable para empezar a introducir a los estudiantes en lenguajes usados en ámbitos profesionales y de trabajo.

La CalculadoraV1 es un ejercicio que se hizo con anterioridad en el módulo 1. Simplemente solicite primero a los estudiantes que se desea usar “sumas, restar, multiplicar o dividir” y después se ingresan los datos a calcular con su respectiva visualización de los datos y el resultado final.

La CalculadoraV2 es una mejora utilizando todas las herramientas necesarias para así poder expresar una mayor magnitud de resultados, con el fin de no solo elegir simplemente dos valores a calcular, sino que para que el usuario decida cuántos valores planea calcular. En este caso, el máximo es de 10 y termina haciendo el cálculo esperado.

**Actitudes:** En esta actividad es recomendable el uso de computadores para codificar elseudocódigo para verificar de una manera eficiente lo realizado, ejecutar y ver el funcionamiento de este. Para el desarrollo del trabajo individual es necesario tener claro que esto es para que el estudiante se sienta seguro de lo aprendido y tener procesos metacognitivos que les permita evaluar su forma de pensar algorítmicamente.

**Orientaciones para organizar e implementar el juego de roles:** Se sugieren las siguientes indicaciones para guiar la implementación del trabajo individual:

- Elaborar un ejercicio simple que pueda ser mejorado de diferentes maneras.
- Determinar un tiempo para así resolver cualquier duda del problema.
- Determinar el tiempo que cree necesario para haber terminado o al menos llevar algo de desarrollo del ejercicio.
- Determinar el tiempo para retroalimentar a los estudiantes.

## Actividad de desempeño 3

### Propósito

Esta actividad se enfocará en las reglas básicas del funcionamiento de “consulta de base de datos”, poniendo el foco en aprovechar oportunidades cotidianas para la creación de tablas, sus restricciones y su respectivo ingreso de información. Asimismo, se seguirá favoreciendo que las soluciones a problemas estén en los dominios del uso de redes e internet, así como en la relación que tienen las personas con los computadores para obtener diferentes respuesta o resultados.

### Objetivos de Aprendizaje

**OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y Transferir)**

### Conocimiento esencial

Redes e Internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.

### Tiempo estimado


3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

Se sugiere que el docente presente a los estudiantes una situación laboral de registro de datos de empleadores según una tabla con ciertas condiciones de entrada. Esta creación de tablas permite organizar los datos de los empleados de forma ordenada y de fácil acceso. También, con estos programas es posible acceder a parte de la información según se requiera.

**En la oficina  
de  
Recursos Humanos**



Hola, bienvenido a la empresa. Le asignaremos un código de 4 caracteres que usaremos de forma interna.

Por favor, dígame su nombre completo....

¿Cuál es su dirección?

En su contrato, el sueldo indicado es...

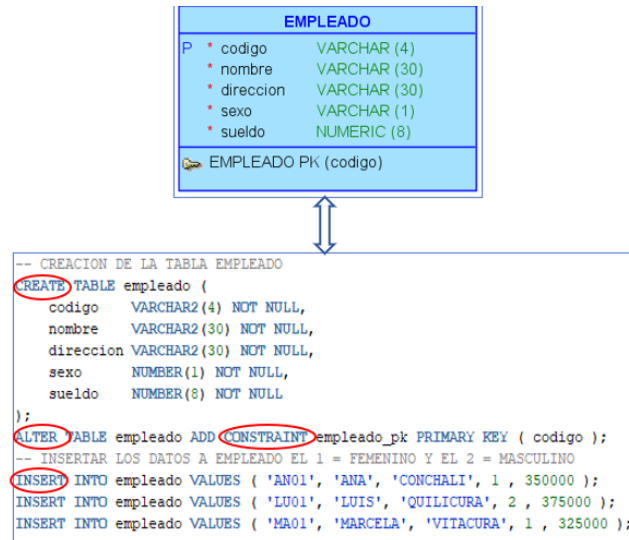
¿En qué sección trabajará?

Algunas de las preguntas que pueden motivar la introducción de las nuevas funciones de creación “CREATE”, de variación “Alter”, de condiciones “CONSTRAINT” y de ingresar “INSERT”, pueden ser:

- ¿Cómo se pueden organizar estos datos?
- ¿Cómo podemos programar la creación de una tabla?
- ¿Qué ocurre cuando se tienen varios empleados?

### Construcción del conocimiento

Se sugiere ampliar el conocimiento sobre las funciones de programación, incluyendo la creación, la variación, el condicional y el ingreso. Para esto, continuar con la misma situación y explicar la transferencia desde el modelo relación-entidad a la programación en pseudocódigo, indicando cada vez lo que hace cada función.



### Práctica guiada

Se sugiere realizar en conjunto el desarrollo del siguiente modelo de base de datos y agregar datos a cada tabla para luego revisar el posible orden de eliminación de datos.

### Modelo entidad-relación

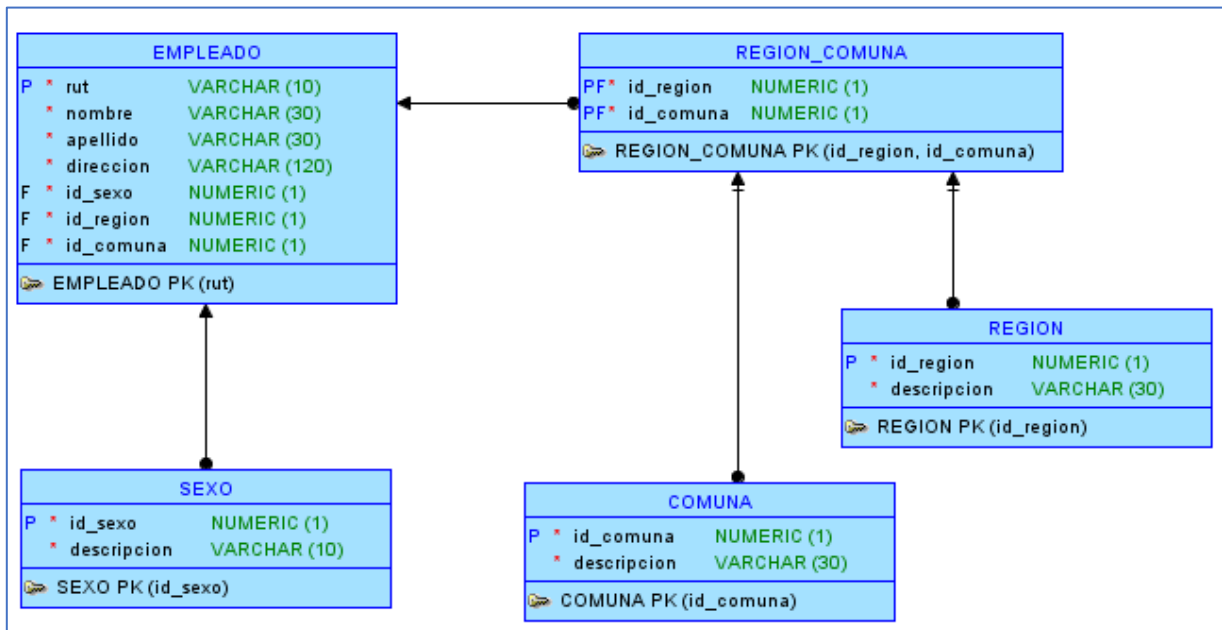


Tabla de datos para ingresar

RUT	NOMBRE	APELLIDO	DIRECCION	ID_SEXO	ID_REGION	ID_COMUNA
18789456-K	Alan	Albornoz	Av Siempre Vida #1234	2	1	1
12321987-9	Xavier	Rojas	Lo Ovalle #659	2	2	4
14852963-7	Macarena	Diaz	Camilo Henriquez #8492	1	3	7

ID_COMUNA	DESCRIPCION
1	Camarones
2	Putre
3	Conchali
4	Quilicura
5	Vitacura
6	Los Lagos
7	Paillaco

ID_REGION	DESCRIPCION
1	ARICA Y PARINACOTA
2	METROPOLITANA DE SANTIAGO
3	LOS RÍOS

ID_SEXO	DESCRIPCION
1	Femenino
2	Masculino
3	Otros...

ID_REGION	ID_COMUNA
1	1
1	2
2	3
2	4
2	5
3	6
3	7

Acción	Resultado esperado	Preguntas de apoyo
<p>Crear tablas con sus respectivas claves primarias y claves foráneas.</p>	<pre>CREATE TABLE comuna (   id_comuna NUMBER(1) NOT NULL CONSTRAINT comuna_pk PRIMARY KEY,   descripcion VARCHAR2(30) NOT NULL);  CREATE TABLE region (   id_region NUMBER(1) NOT NULL CONSTRAINT region_pk PRIMARY KEY,   descripcion VARCHAR2(30) NOT NULL);  CREATE TABLE region_comuna (   id_region NUMBER(1) NOT NULL,   id_comuna NUMBER(1) NOT NULL,   CONSTRAINT region_comuna_pk PRIMARY KEY ( id_region, id_comuna ),   CONSTRAINT reg_com_comuna_fk FOREIGN KEY ( id_comuna )   REFERENCES comuna ( id_comuna ),   CONSTRAINT reg_com_region_fk FOREIGN KEY ( id_region )   REFERENCES region ( id_region ));  CREATE TABLE sexo (   id_sexo NUMBER(1) NOT NULL CONSTRAINT sexo_pk PRIMARY KEY,   descripcion VARCHAR2(10) NOT NULL);  CREATE TABLE empleado (   rut VARCHAR2(10) NOT NULL CONSTRAINT empleado_pk PRIMARY KEY,   nombre VARCHAR2(30) NOT NULL,   apellido VARCHAR2(30) NOT NULL,   direccion VARCHAR2(120) NOT NULL,   id_sexo NUMBER(1) NOT NULL,   id_region NUMBER(1) NOT NULL,   id_comuna NUMBER(1) NOT NULL,   CONSTRAINT empleado_reg_com_fk FOREIGN KEY ( id_region, id_comuna )   REFERENCES region_comuna ( id_region, id_comuna ),   CONSTRAINT empleado_sexo_fk FOREIGN KEY ( id_sexo )   REFERENCES sexo ( id_sexo ));</pre>	<p>¿Cómo saber con cuál empezar?</p>

Acción	Resultado esperado	Preguntas de apoyo
Elaborar el ingreso de datos a las tablas anteriormente ya creadas.	<pre>INSERT INTO sexo VALUES (1,'Femenino'); INSERT INTO sexo VALUES (2,'Masculino'); INSERT INTO sexo VALUES (3,'Otros...');  INSERT INTO comuna VALUES (1,'Camarones'); INSERT INTO comuna VALUES (2,'Putre'); INSERT INTO comuna VALUES (3,'Conchali'); INSERT INTO comuna VALUES (4,'Quilicura'); INSERT INTO comuna VALUES (5,'Vitacura'); INSERT INTO comuna VALUES (6,'Los Lagos'); INSERT INTO comuna VALUES (7,'Paillaco');  INSERT INTO region VALUES (1,'ARICA Y PARINACOTA'); INSERT INTO region VALUES (2,'METROPOLITANA DE SANTIAGO'); INSERT INTO region VALUES (3,'LOS RÍOS');  INSERT INTO region_comuna VALUES (1,1); INSERT INTO region_comuna VALUES (1,2); INSERT INTO region_comuna VALUES (2,3); INSERT INTO region_comuna VALUES (2,4); INSERT INTO region_comuna VALUES (2,5); INSERT INTO region_comuna VALUES (3,6); INSERT INTO region_comuna VALUES (3,7);  INSERT INTO empleado VALUES ( '18789456-K','Alan','Albornoz', 'Av Siempre Vida #1234',2,1,1); INSERT INTO empleado VALUES ( '12321987-9','Xavier','Rojas', 'Lo Ovalle #659', 2,2,4); INSERT INTO empleado VALUES ( '14852963-7','Macarena','Diaz', 'Camilo Henriquez #8492',1,3,7);</pre>	¿En qué orden debería insertar los datos?
Elaborar la eliminación de las tablas.	<pre>DROP TABLE empleado; DROP TABLE sexo; DROP TABLE region_comuna; DROP TABLE comuna; DROP TABLE region;</pre>	¿En qué orden deberían eliminarse?

### Práctica independiente

Se sugiere un trabajo de pares y continuar con el ejercicio anterior, enfatizando una tarea particular, por ejemplo: la gerencia de recursos humanos requiere contar con una lista de empleados donde esté toda la información necesaria y ordenada en orden alfabético. La lista debe tener el formato y orden que se aprecia en la imagen.

RUT	NOMBRE COMPLETO	DIRECCION	SEXO	COMUNA	REGION
18789456-K	Alan Albornoz	Av Siempre Vida #1234	Masculino	Camarones	ARICA Y PARINACOTA
14852963-7	Macarena Diaz	Camilo Henriquez #8492	Femenino	Paillaco	LOS RÍOS
12321987-9	Xavier Rojas	Lo Ovalle #659	Masculino	Quilicura	METROPOLITANA DE SANTIAGO

Acción	Resultado esperado
Realizar un SELECT FROM para poder	<pre>SELECT RUT, NOMBRE, APELLIDO, DIRECCION, ID_SEXO, ID_COMUNA, ID_REGION FROM EMPLEADO;</pre>



Acción	Resultado esperado																												
visualizar los datos de EMPLEADO.	<table border="1"> <thead> <tr> <th>RUT</th> <th>NOMBRE</th> <th>APELLIDO</th> <th>DIRECCION</th> <th>ID_SEXO</th> <th>ID_COMUNA</th> <th>ID_REGION</th> </tr> </thead> <tbody> <tr> <td>18789456-K</td> <td>Alan</td> <td>Albornoz</td> <td>Av Siempre Vida #1234</td> <td>2</td> <td>1</td> <td>1</td> </tr> <tr> <td>12321987-9</td> <td>Xavier</td> <td>Rojas</td> <td>Lo Ovalle #659</td> <td>2</td> <td>4</td> <td>2</td> </tr> <tr> <td>14852963-7</td> <td>Macarena</td> <td>Diaz</td> <td>Camilo Henriquez #8492</td> <td>1</td> <td>7</td> <td>3</td> </tr> </tbody> </table>	RUT	NOMBRE	APELLIDO	DIRECCION	ID_SEXO	ID_COMUNA	ID_REGION	18789456-K	Alan	Albornoz	Av Siempre Vida #1234	2	1	1	12321987-9	Xavier	Rojas	Lo Ovalle #659	2	4	2	14852963-7	Macarena	Diaz	Camilo Henriquez #8492	1	7	3
RUT	NOMBRE	APELLIDO	DIRECCION	ID_SEXO	ID_COMUNA	ID_REGION																							
18789456-K	Alan	Albornoz	Av Siempre Vida #1234	2	1	1																							
12321987-9	Xavier	Rojas	Lo Ovalle #659	2	4	2																							
14852963-7	Macarena	Diaz	Camilo Henriquez #8492	1	7	3																							
Realizar un SELECT FROM + JOIN para llamar las tablas que están relacionadas con la tabla EMPLEADO.	<pre>SELECT E.RUT, E.NOMBRE, E.APELLIDO, E.DIRECCION, S.ID_SEXO, C.ID_COMUNA, R.ID_REGION FROM EMPLEADO E JOIN COMUNA C ON C.ID_COMUNA = E.ID_COMUNA JOIN REGION R ON R.ID_REGION = E.ID_REGION JOIN SEXO S ON S.ID_SEXO = E.ID_SEXO;</pre>																												
Realizar los ajustes finales (llamar la descripción de las tablas para poner el nombre y no el ID) y darles un orden dependiendo del nombre del empleado.	<pre>SELECT RUT, NOMBRE    ' '    APELLIDO "NOMBRE COMPLETO", DIRECCION, S.DESCRIPCION SEXO, C.DESCRIPCION COMUNA, R.DESCRIPCION REGION FROM EMPLEADO E JOIN COMUNA C ON C.ID_COMUNA = E.ID_COMUNA JOIN REGION R ON R.ID_REGION = E.ID_REGION JOIN SEXO S ON S.ID_SEXO = E.ID_SEXO ORDER BY E.NOMBRE;</pre>																												

### Evaluación formativa

Para verificar aprendizajes, se sugiere utilizar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Creación de tablas	Crea tablas, pero no se ejecuta la secuencia.	Crea tablas, pero debe repetirse el proceso para su ejecución.	Crea tablas que ejecutan la secuencia sin errores.
Inserción de tablas	Inserta tablas, pero no se ejecuta la secuencia	Inserta tablas, pero debe repetirse el proceso para su ejecución.	Inserta tablas que ejecutan la secuencia sin errores.
Eliminación de tablas	Elimina tablas, pero no se ejecuta la secuencia	Elimina tablas, pero debe repetirse el proceso para su ejecución.	Elimina tablas que ejecutan la secuencia sin errores.

## Orientaciones al docente

**Para unificar conceptos disciplinares:** en esta actividad se deja de lado el modelamiento de base de datos, poniendo énfasis en la consulta y creación de base de datos, las aplicaciones utilizadas, como Oracle DataModeler y se utiliza además Oracle Database Express Edition (XE) Release (18c) y Oracle SQL Developer 21.2.1.

Los comandos tienen como objetivo permitir la manipulación de los datos existentes en las tablas de la base de datos:

- **CREATE:** Permite crear tablas.
- **ALTER:** Permite alterar tablas
- **DROP:** Permite eliminar tablas.
- **INSERT:** Permite añadir filas a una tabla.
- **UPDATE:** Permite actualizar valores de una o más columnas para un subconjunto de las filas de una tabla.
- **DELETE:** Borra una o más filas de una tabla, dependiendo de la condición WHERE o de las dependencias de restricciones de clave foránea.
- **SELECT:** Permite la selección de datos sobre la tabla. Consiste en elegir un subconjunto de filas que cumplan con algunas condiciones determinadas. También se pueden seleccionar todas las filas de una tabla.
- **COMMIT:** Guardar los cambios de la transacción en curso (para guardar los datos insertados).
- **ROLLBACK:** deshace los cambios de la transacción en curso (para cancelar la inserción de datos.)

Las condiciones o CONSTRAINTS que afectan a una tabla son las siguientes:

- **NOT NULL:** Especifica que la columna no puede almacenar valores nulos (vacíos).
- **UNIQUE:** Especifica que el valor de la columna debe ser único para todas las filas en la tabla. No puede ser referenciada como foránea.
- **PRIMARY KEY:** Identifica en forma única cada fila de la tabla y puede ser referenciada como foránea.
- **FOREIGN KEY (FK):** Establece y hace cumplir una relación de clave foránea entre la columna de la tabla a crear y una columna de la tabla referenciada.
- **CHECK:** Especifica una condición que debe ser verdadera para los valores que almacenará la columna.

Se sugiere comenzar con ejemplos sencillos de cada nueva función. Un ejemplo de condiciones podría ser:

Vendedor			
N° (VNR)	Nombre (Vnombre)	Posición	Sueldo (U\$)
2011	Reyes	Junior	1 000
2012	Espinoza	Senior	2 000

Cliente		
N° (CNR)	Nombre (Cnombre)	Supervisor
1	Astudillo	2011
2	Martí	2012



Marco para programar en pseudocódigo:

- El número del cliente es una llave primaria.
- El cliente tiene nombre.
- El supervisor es una llave foránea para conectar la tabla del vendedor.

Posible programación para completar en pseudocódigo:

```
CREATE TABLE Cliente (
    CNR NÚMERO,
    Nombre VARCHAR (30)
    CONSTRAINT Cnombre NOT NULL,
    Supervisor NÚMERO,
    PRIMARY KEY (CNR),
    CONSTRAINT FK_Cliente
    FOREIGN KEY (Supervisor)
    REFERENCES Vendedor (VNR)
```

**Actitudes:** Para apoyar el desarrollo de las actividades es recomendable el uso de computadores, ya que hay que agregar tablas y la información de esas tablas es a nivel informático, por tanto, es recomendable tener el *hardware* y *software* necesario para poder trabajar de la mejor manera posible. En cuanto al desarrollo de la actividad en pares, se sugiere promover la ayuda entre los estudiantes para conseguir resolver los problemas.

**Orientaciones para organizar e implementar el trabajo entre pares:** Se sugieren las siguientes indicaciones para guiar la implementación del trabajo en pares:

- Elaborar un modelo de base de datos de al menos 10 y máximo 20 tablas.
- Ambos estudiantes tendrán que crear esas tablas mediante un Script en base al modelo que tendrán.
- Determinar el tiempo necesario para elaborar el Script.
- Determine lo que se hará con los resultados. Si es necesario puede mostrar el Script de algún estudiante y revisarlo junto a la clase para analizar los problemas si es que los hay y los aciertos del problema. Se sugiere conversar sobre las dudas que pueden llegar a haber en cuanto a la creación de estos Scripts.

## Actividad de desempeño 4

### Propósito

Esta actividad pone el foco en el mundo de la consulta de base de datos, transfiriendo los conocimientos y habilidades para el manejo de las herramientas que se pueden utilizar para trabajar con estructuras informacionales complejas y precisas para llegar a conseguir y aplicar soluciones flexiblemente a diferentes problemas.

### Objetivos de Aprendizaje

**OA4.** Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. **(Generalizar y Transferir)**

**OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista. **(Generalizar y Transferir)**

### Conocimiento esencial

Redes e Internet: sitios web, aplicaciones móviles, seguridad cibernética y objetos controlados digitalmente.

### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

Se sugiere que el docente presente dos tablas iguales, considerando una sin ninguna modificación y otra que está consultando datos precisos, complementarios y organizados:

EMPLOYEE_ID	NAME	EMAIL	JOB_TITLE	SALARY	MANAGER_ID	DEPARTMENT_NAME
1	100 Steven King	SKING	President	\$24,000.00	Sin Jefe	Executive
2	101 Neena Kochhar	NKOCHHAR	Administration Vice President	\$17,000.00	100	Executive
3	102 Lex De Haan	LDEHAAN	Administration Vice President	\$17,000.00	100	Executive
4	103 Alexander Hunold	AHUNOLD	Programmer	\$9,000.00	102	IT
5	104 Bruce Ernst	BERNST	Programmer	\$6,000.00	103	IT
6	105 David Austin	DAUSTIN	Programmer	\$4,800.00	103	IT
7	106 Valli Pataballa	VPATABAL	Programmer	\$4,800.00	103	IT
8	107 Diana Lorentz	DLORENTZ	Programmer	\$4,200.00	103	IT
9	108 Nancy Greenberg	NGREENBE	Finance Manager	\$12,008.00	101	Finance
10	109 Daniel Faviet	DFAVIET	Accountant	\$9,000.00	108	Finance

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
1	100 Steven	King	SKING	AD_PRES	24000	(null)	90
2	101 Neena	Kochhar	NKOCHHAR	AD_VP	17000	100	90
3	102 Lex	De Haan	LDEHAAN	AD_VP	17000	100	90
4	103 Alexander	Hunold	AHUNOLD	IT_PROG	9000	102	60
5	104 Bruce	Ernst	BERNST	IT_PROG	6000	103	60
6	105 David	Austin	DAUSTIN	IT_PROG	4800	103	60
7	106 Valli	Pataballa	VPATABAL	IT_PROG	4800	103	60
8	107 Diana	Lorentz	DLORENTZ	IT_PROG	4200	103	60
9	108 Nancy	Greenberg	NGREENBE	FI_MGR	12008	101	100
10	109 Daniel	Faviet	DFAVIET	FI_ACCOUNT	9000	108	100

Algunas de las preguntas que podrían promover procesos para distinguir la consulta de bases de datos podrían ser:

- ¿Para qué sirve la consulta de base de datos?
- ¿Cuál es la funcionalidad de ofrecer datos más específicos al usuario?

### Construcción de conocimiento

Se sugiere que el docente presente a los estudiantes un desafío en un contexto real en donde utilizando la base de datos "HR" se necesiten las siguientes especificaciones para esta consulta de base de datos, por ejemplo:

- Ordenar los datos dependiendo del nuevo sueldo de forma descendente.
- Mostrar el nombre y el apellido juntos.
- Hacer un nuevo salario dependiendo de los siguientes datos:
  - Si trabaja en "PR\_REP" su salario será aumentado el 5%
  - Si trabaja en "SA\_MAN" su salario será aumentado el 3 %
  - Si trabaja en "IT\_PROG" su salario será aumentado un 1%

- Hacer un ranking dependiendo del nuevo salario:
  - Si salario es inferior a 5 000 dólares, es un salario bajo.
  - Si salario está entre 5 000 y 10 000 dólares, es un salario medio.
  - Si salario es superior a 10 000 dólares, es un salario alto.
- La fecha debe tener el formato que se ve en la imagen.

EMPLOYEE_ID	NAME	JOB_ID	SALARY	NEW_SALARY	RANKING	MANAGER_ID	HIRE_DATE
1	145 John Russell	SA_MAN	\$14,000.00	\$14,420.00	ALTO	100	01 de octubre del 2004
2	146 Karen Partners	SA_MAN	\$13,500.00	\$13,905.00	ALTO	100	05 de enero del 2005
3	147 Alberto Errazuriz	SA_MAN	\$12,000.00	\$12,360.00	ALTO	100	10 de marzo del 2005
4	148 Gerald Cambrault	SA_MAN	\$11,000.00	\$11,330.00	ALTO	100	15 de octubre del 2007
5	149 Eleni Zlotkey	SA_MAN	\$10,500.00	\$10,815.00	ALTO	100	29 de enero del 2008
6	204 Hermann Baer	PR_REP	\$10,000.00	\$10,500.00	ALTO	101	07 de junio del 2002
7	103 Alexander Hunold	IT_PROG	\$9,000.00	\$9,900.00	MEDIO	102	03 de enero del 2006
8	104 Bruce Ernst	IT_PROG	\$6,000.00	\$6,600.00	MEDIO	103	21 de mayo del 2007
9	106 Valli Pataballa	IT_PROG	\$4,800.00	\$5,280.00	MEDIO	103	05 de febrero del 2006
10	105 David Austin	IT_PROG	\$4,800.00	\$5,280.00	MEDIO	103	25 de junio del 2005
11	107 Diana Lorentz	IT_PROG	\$4,200.00	\$4,620.00	BAJO	103	07 de febrero del 2007

Acción	Resultado esperado	Preguntas de apoyo
Organizar datos y primer avance de la consulta	<pre>SELECT employee_id, first_name, last_name, job_ID, salary, manager_id, department_ID, hire_date FROM EMPLOYEES WHERE JOB_ID = 'PR_REP' OR JOB_ID = 'SA_MAN' OR JOB_ID = 'IT_PROG';</pre>	¿Por dónde empezar?
Desarrollar el “WHEN - CASE” relacionado con la dependencia del trabajo para identificar el aumento del trabajo	<pre>SELECT employee_id, first_name, last_name, job_ID, salary, CASE JOB_ID WHEN 'PR_REP' THEN SALARY+SALARY*0.05 WHEN 'SA_MAN' THEN SALARY+SALARY*0.03 WHEN 'IT_PROG' THEN SALARY+SALARY*0.1 ELSE SALARY END NEW_SALARY, manager_id, department_ID, hire_date FROM EMPLOYEES WHERE JOB_ID = 'PR_REP' OR JOB_ID = 'SA_MAN' OR JOB_ID = 'IT_PROG';</pre>	

Acción	Resultado esperado	Preguntas de apoyo
<p>Desarrollar el segundo “WHEN - CASE” para la clasificación del nuevo salario</p>	<pre>SELECT employee_id, first_name, last_name, job_ID, salary, CASE JOB_ID WHEN 'PR_REP' THEN SALARY+SALARY*0.05 WHEN 'SA_MAN' THEN SALARY+SALARY*0.03 WHEN 'IT_PROG' THEN SALARY+SALARY*0.1 ELSE SALARY END NEW_SALARY, CASE WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) &lt; 5000 THEN 'BAJO' WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) BETWEEN 5000 AND 10000 THEN 'MEDIO' WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) &gt; 10000 THEN 'ALTO' ELSE ' ' END RANKING, manager_id, department_ID, hire_date FROM EMPLOYEES WHERE JOB_ID = 'PR_REP' OR JOB_ID = 'SA_MAN' OR JOB_ID = 'IT_PROG';</pre>	<p>¿Cómo clasifico el nuevo salario?</p>
<p>Últimos detalles y finalización de la consulta</p>	<pre>SELECT employee_id, first_name    ' '    last_name NAME, JOB_ID, TO_CHAR (SALARY, '\$99,999.00') SALARY, CASE JOB_ID WHEN 'PR_REP' THEN TO_CHAR ( SALARY+SALARY*0.05, '\$99,999.00') WHEN 'SA_MAN' THEN TO_CHAR (SALARY+SALARY*0.03, '\$99,999.00') WHEN 'IT_PROG' THEN TO_CHAR (SALARY+SALARY*0.1, '\$99,999.00') ELSE TO_CHAR (SALARY, '\$99,999.00') END NEW_SALARY, CASE WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) &lt; 5000 THEN 'BAJO' WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) BETWEEN 5000 AND 10000 THEN 'MEDIO' WHEN DECODE(JOB_ID, 'PR_REP', SALARY+SALARY*0.05, 'SA_MAN', SALARY+SALARY*0.03, 'IT_PROG', SALARY+SALARY*0.1, SALARY) &gt; 10000 THEN 'ALTO' ELSE ' ' END RANKING, NVL (TO_CHAR(MANAGER_ID), 'SIN JEFE') MANAGER_ID, TO_CHAR (HIRE_DATE, 'dd "de" month "del" yyyy') HIRE_DATE FROM EMPLOYEES WHERE JOB_ID = 'PR_REP' OR JOB_ID = 'SA_MAN' OR JOB_ID = 'IT_PROG' ORDER BY NEW_SALARY DESC;</pre>	

### Práctica guiada

En esta práctica, se sugiere que el docente haga varias demostraciones de cómo usar ciertos elementos de la consulta de base de datos en los cuales priorizan el uso del JOIN:

Acción	Resultado esperado																																																
Mostrar para cada empleado en qué área de trabajo se desempeña, el departamento correspondiente y la ciudad en la que se encuentra.	<pre>SELECT FIRST_NAME, LAST_NAME, D.DEPARTMENT_NAME, J.JOB_TITLE, J.MAX_SALARY, E.SALARY, L.CITY, L.COUNTRY_ID FROM EMPLOYEES E JOIN DEPARTMENTS D ON D.DEPARTMENT_ID=E.DEPARTMENT_ID JOIN JOBS J ON J.JOB_ID=E.JOB_ID JOIN LOCATIONS L ON L.LOCATION_ID=D.LOCATION_ID;</pre>																																																
Resultado esperado	<table border="1"> <thead> <tr> <th>FIRST_NAME</th> <th>LAST_NAME</th> <th>DEPARTMENT_NAME</th> <th>JOB_TITLE</th> <th>MAX_SALARY</th> <th>SALARY</th> <th>CITY</th> <th>COUNTRY_ID</th> </tr> </thead> <tbody> <tr> <td>Steven</td> <td>King</td> <td>Executive</td> <td>President</td> <td>40000</td> <td>24000</td> <td>Seattle</td> <td>US</td> </tr> <tr> <td>Neena</td> <td>Kochhar</td> <td>Executive</td> <td>Administration Vice President</td> <td>30000</td> <td>17000</td> <td>Seattle</td> <td>US</td> </tr> <tr> <td>Lex</td> <td>De Haan</td> <td>Executive</td> <td>Administration Vice President</td> <td>30000</td> <td>17000</td> <td>Seattle</td> <td>US</td> </tr> <tr> <td>Jennifer</td> <td>Whalen</td> <td>Administration</td> <td>Administration Assistant</td> <td>6000</td> <td>4400</td> <td>Seattle</td> <td>US</td> </tr> <tr> <td>Nancy</td> <td>Greenberg</td> <td>Finance</td> <td>Finance Manager</td> <td>16000</td> <td>12008</td> <td>Seattle</td> <td>US</td> </tr> </tbody> </table>	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	JOB_TITLE	MAX_SALARY	SALARY	CITY	COUNTRY_ID	Steven	King	Executive	President	40000	24000	Seattle	US	Neena	Kochhar	Executive	Administration Vice President	30000	17000	Seattle	US	Lex	De Haan	Executive	Administration Vice President	30000	17000	Seattle	US	Jennifer	Whalen	Administration	Administration Assistant	6000	4400	Seattle	US	Nancy	Greenberg	Finance	Finance Manager	16000	12008	Seattle	US
FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	JOB_TITLE	MAX_SALARY	SALARY	CITY	COUNTRY_ID																																										
Steven	King	Executive	President	40000	24000	Seattle	US																																										
Neena	Kochhar	Executive	Administration Vice President	30000	17000	Seattle	US																																										
Lex	De Haan	Executive	Administration Vice President	30000	17000	Seattle	US																																										
Jennifer	Whalen	Administration	Administration Assistant	6000	4400	Seattle	US																																										
Nancy	Greenberg	Finance	Finance Manager	16000	12008	Seattle	US																																										
Mostrar los departamentos que no tienen empleados	<pre>SELECT FIRST_NAME, LAST_NAME, D.DEPARTMENT_NAME FROM EMPLOYEES E RIGHT OUTER JOIN DEPARTMENTS D ON D.DEPARTMENT_ID=E.DEPARTMENT_ID WHERE E.EMPLOYEE_ID IS NULL;</pre>																																																
Resultado esperado	<table border="1"> <thead> <tr> <th>FIRST_NAME</th> <th>LAST_NAME</th> <th>DEPARTMENT_NAME</th> </tr> </thead> <tbody> <tr> <td>(null)</td> <td>(null)</td> <td>Treasury</td> </tr> <tr> <td>(null)</td> <td>(null)</td> <td>Corporate Tax</td> </tr> <tr> <td>(null)</td> <td>(null)</td> <td>Control And Credit</td> </tr> <tr> <td>(null)</td> <td>(null)</td> <td>Shareholder Services</td> </tr> </tbody> </table>	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	(null)	(null)	Treasury	(null)	(null)	Corporate Tax	(null)	(null)	Control And Credit	(null)	(null)	Shareholder Services																																	
FIRST_NAME	LAST_NAME	DEPARTMENT_NAME																																															
(null)	(null)	Treasury																																															
(null)	(null)	Corporate Tax																																															
(null)	(null)	Control And Credit																																															
(null)	(null)	Shareholder Services																																															
Agregar la tabla rango de sueldo y trabajar mediante estos datos	<pre>CREATE TABLE RANGO_SUELDO( ID_RS INTEGER NOT NULL PRIMARY KEY, RANGO_MIN NUMBER, RANGO_MAX NUMBER, DESCRIPCION VARCHAR2(50) );  INSERT INTO RANGO_SUELDO VALUES (1,0,3999,'BAJO'); INSERT INTO RANGO_SUELDO VALUES (2,4000,6999,'MEDIO BAJO'); INSERT INTO RANGO_SUELDO VALUES (3,7000,9999,'MEDIO'); INSERT INTO RANGO_SUELDO VALUES (4,10000,40000,'ALTO');  SELECT FIRST_NAME, LAST_NAME, SALARY, RS.DESCRIPCION FROM EMPLOYEES E JOIN RANGO_SUELDO RS ON E.SALARY BETWEEN RS.RANGO_MIN AND RS.RANGO_MAX ORDER BY E.EMPLOYEE_ID;</pre>																																																



Resultado esperado	FIRST_NAME	LAST_NAME	SALARY	DESCRIPCION
	Steven	King	24000	ALTO
	Neena	Kochhar	17000	ALTO
	Lex	De Haan	17000	ALTO
	Alexander	Hunold	9000	MEDIO
	Bruce	Ernst	6000	MEDIO BAJO
	David	Austin	4800	MEDIO BAJO
	Valli	Pataballa	4800	MEDIO BAJO

### Práctica independiente

Se recomienda al docente favorecer un trabajo individual en donde se ejercite un desafío en un contexto real, en donde utilizando base de datos se requieran especificaciones para esta consulta de base de datos. También, se puede ocupar en muchos trabajos donde se deba hacer una o dos cosas para completar la consulta (se sugiere trabajar en una base de datos nueva con nuevas tablas e información). Por ejemplo: se puede trabajar en mostrar solamente datos de quiénes nacieron en algún mes o año, también se podría trabajar con darle un formato numérico en específico o agregar nuevas tablas, etc.

### Evaluación formativa

Para verificar aprendizajes, se sugiere utilizar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Consulta de base de datos	Genera Script.	Genera Script de manera correcta, pero algunos datos no se muestran cómo deberían.	Genera Script con todo lo solicitado dentro del problema.
Uso de herramientas	Usa herramientas que no son necesarias.	Usa las herramientas que son necesarias, pero de manera ineficiente.	Usa las herramientas necesarias y consigue generar un Script eficiente.

### Orientaciones al docente

**Para unificar conceptos disciplinares:** Es posible considerar que la consulta de base de datos es muy específica al momento de hacer ejercicios y, por lo tanto, siempre se debe enseñar cómo funcionan los comandos de manera que el estudiante entienda el concepto principal y el uso que se les puede dar. Se está trabajando con la base de datos de Oracle, específicamente la de recursos humanos “HR”, esto debido a que es una base de datos que se puede desbloquear y probar siempre que se use Oracle. A partir de estas herramientas, se puede tener una base sólida para trabajar con las demás herramientas que necesitan más conceptualización y, a su vez, son de una dificultad elevada para entender todas las funcionalidades u ocasiones en donde se podrían utilizar.

**Actitudes:** Para apoyar el desarrollo de las actividades, se recomienda el uso de computadores, ya que trabajos con información precisa debe ser evaluada constantemente para poder apreciar los cambios y errores que puedan llegar a ocurrir. En cuanto al desarrollo de la actividad individual, es para favorecer al estudiante para que tenga autonomía y seguridad para resolver los ejercicios en contextos de aplicación real.

**Orientaciones para organizar e implementar el trabajo individual:** Se sugieren las siguientes indicaciones para guiar al estudiante en el trabajo individual.

- Elaborar un ejercicio amplio como en la construcción de conocimiento o implementar varios ejercicios con dificultad moderada en el cual se le solicite entre uno a dos cambios.
- Determinar un tiempo específico para poder realizar las actividades.
- Dar retroalimentación del ejercicio.
- Favorecer la reflexión, evaluación y generalizar los aprendido durante la actividad y módulo.

## Módulo Obligatorio 4

### Visión panorámica

#### Gran idea

El pensamiento algorítmico permite organizar pasos y reglas para diseñar soluciones y resolver una diversidad de problemas cuya comprensión se favorece al integrar conceptos matemáticos como variables, funciones, ecuaciones o gráficos, entre otros.

#### Objetivos de aprendizaje

**OA4.** Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada.

**(Generalizar y Transferir)**

**OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas en otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista.

**(Generalizar y Transferir)**

**OA6.** Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

#### Conocimientos esenciales

- Algoritmos y programación: lenguajes, métodos y entornos de programación.
- Representación y análisis de datos.
- Impacto social de la computación: interacciones, cultura, seguridad y ética.

#### Tiempo estimado

6 semanas (12 horas)

## Propósito Módulo obligatorio 4

En el módulo 4 de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media se espera que los estudiantes puedan pensar de forma algorítmica para organizar los pasos que les permitan resolver uno o más problemas, integrando conceptos matemáticos tales como variables, funciones, ecuaciones o gráficos. Asimismo, se favorece la capacidad de evaluar y seleccionar algoritmos en función de su rendimiento, reutilización y la facilidad de implementarlos, considerando que un algoritmo óptimo es aquel que es eficiente en trabajar, economiza tiempo y reduce la redundancia, indistintamente del lenguaje de programación que se utilice.

Sobre este escenario, el módulo 4 pone mayor énfasis en el desarrollo del pensamiento algorítmico a través de procesos de generalización y transferencia de soluciones a problemas que requieran el uso de algoritmos eficientes al momento de ejecutarlos o de requerir una mayor eficiencia y optimización de su estructura. Esto permitirá a los estudiantes utilizar las bases del algoritmo para crear códigos funcionales dentro de un lenguaje de programación. Se espera que los estudiantes tengan experiencias en diferentes entornos de programación al momento de analizar y representar datos en un ambiente digital y avancen hacia la comprensión del impacto social que tiene la computación en las interacciones de las personas a nivel local y global, en la cultura, en la seguridad y en el uso ético de la misma.

Los Objetivos de Aprendizaje del módulo desarrollan las actitudes del siglo XXI del ámbito Maneras de Trabajar y Herramientas para trabajar, promoviendo el uso provechoso de las herramientas para dar soluciones a problemáticas cotidianas en ambientes digitales y actuando de acuerdo con principios éticos. Asimismo, el conocimiento asociado a la alfabetización en tecnologías digitales, desde la perspectiva de la programación, permitirá a los estudiantes acceder y manejar información desde distintas fuentes, para poder decidir cuál y cómo usarla. Finalmente, las actividades de desempeño propuestas esperan consolidar procesos metacognitivos en cuanto al pensamiento computacional, el procesamiento de datos y la entrega de soluciones a problemas que existen en el entorno cotidiano.

## Ruta de aprendizaje del Módulo obligatorio 4

¿Por qué las personas evalúan y seleccionan algoritmos en función del rendimiento, reutilización y la facilidad de implementación para resolver un problema?

### Actividad de desempeño

**1:** Evaluar las herramientas necesarias para poder trabajar con información requerida de forma eficiente y precisa.

### Actividad de desempeño

**2:** Implementar algoritmos que permitan trabajar con los datos necesarios para lograr obtener una solución a una problemática específica.



### Actividad de desempeño 3:

Aplicar un proceso lógico que vincule el sistema de un algoritmo con un lenguaje de programación determinando.

### Actividad de desempeño 4:

Elaborar e implementar una variedad de algoritmos para resolver un problema usando un lenguaje de programación profesional.

## Actividad de desempeño 1

### Propósito

Esta actividad desafía a los estudiantes a trabajar mediante la conexión o relación que existen entre las tablas que conforman una base de datos con el fin de extender las herramientas, evaluarlas y aumentar la comprensión de la información almacenada en ellas. Asimismo, los estudiantes deberán trabajar con información requerida de forma eficiente y precisa, aplicando funcionalidades de consulta y representación de datos.

### Objetivos de Aprendizaje

**OA4.** Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada. (**Generalizar y Transferir**)

### Conocimiento esencial

Representación y análisis de datos.

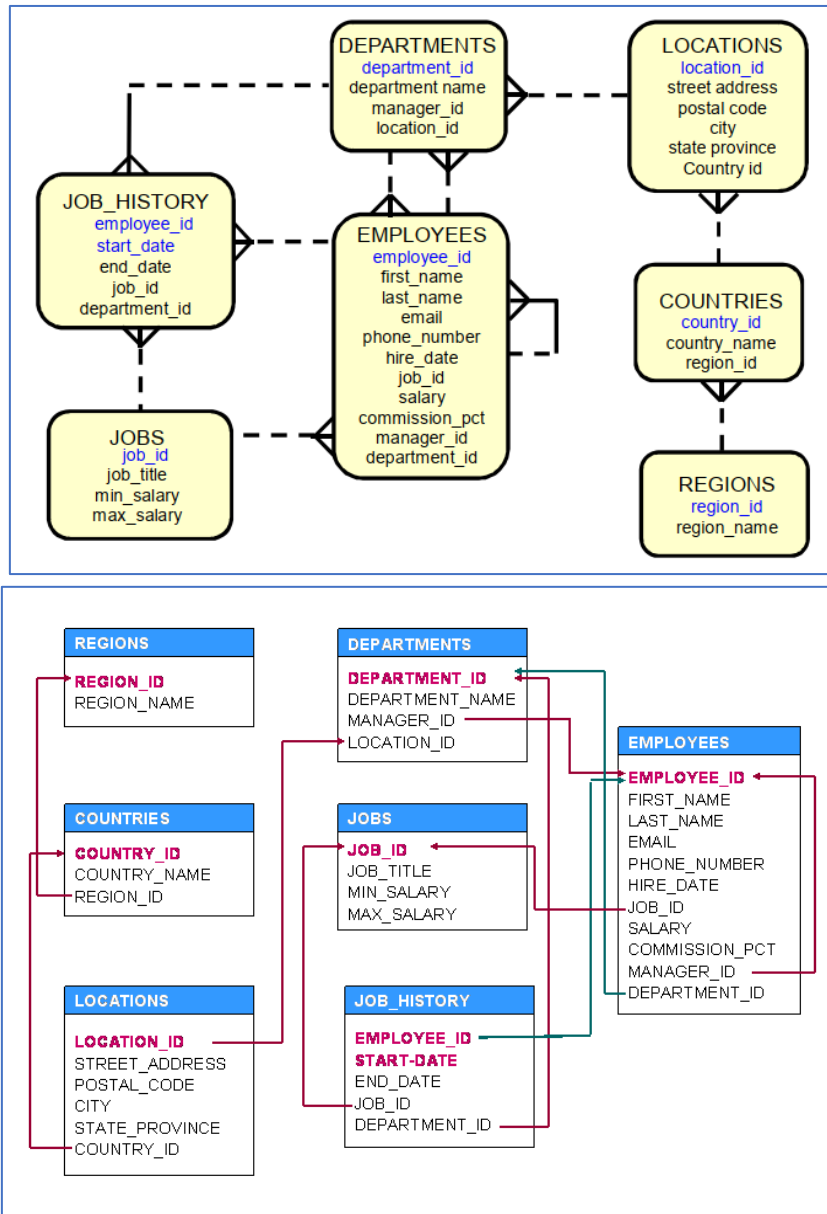
### Tiempo estimado

3 horas pedagógicas.

## Desarrollo de la actividad

### Situación experiencial

Se sugiere que el docente presente dos modelos de datos entidad-relación, en donde ambos den cuenta de las diferentes formas de explicar un modelo de base de datos que proporciona Oracle en su servicio, llamada “HR” (Recursos Humanos):



Algunas de las preguntas que podrían promover procesos de pensamiento asociados a comunicar la importancia de las relaciones entre tablas podrían ser:

- ¿Qué es lo que explica cada modelo?
- ¿Cuál es la importancia de las relaciones en la consulta de base de datos?
- ¿Por qué es útil tener claros los datos de una base de datos?

## Construcción de conocimiento

Para profundizar en el objetivo de la actividad, se sugiere al docente proponer a los estudiantes que, utilizando la base de datos “HR”, se necesitan las siguientes especificaciones para esta consulta de base de datos:

- Mostrar el nombre y el apellido juntos.
- Hacer una descripción de la comisión y darle las opciones:
  - Si es “nulo”, la opción será “Solo sueldo”
  - Si tiene un valor, la opción será “Sueldo + Comisión”
- Se debe mostrar la fecha de contratación y del término del contrato.
- Crear una categoría donde se vean los días, meses y años trabajados.
- El formato de las fechas y el sueldo deben ser el mismo que el de la imagen.
- El resultado final debe mostrar la misma cantidad de datos e información.

ID EMPLEADO	EMPLEADO	SUELDO	COMISION	COMISION DESCRIPCION	CARGO	FECHA DE CONTRATACION	FECHA DE TERMINO	DIAS / MESES / AÑOS CONTRATADOS
1	101 Neena Kochhar	\$17,000	0	Solo Sueldo	Public Accountant	21 de 09 del 1997	27 de 10 del 2001	1497 / 49 / 4
2	101 Neena Kochhar	\$17,000	0	Solo Sueldo	Accounting Manager	28 de 10 del 2001	15 de 03 del 2005	1234 / 41 / 3
3	102 Lex De Haan	\$17,000	0	Solo Sueldo	Programmer	13 de 01 del 2001	24 de 07 del 2006	2018 / 66 / 6
4	114 Den Raphaely	\$11,000	0	Solo Sueldo	Stock Clerk	24 de 03 del 2006	31 de 12 del 2007	647 / 21 / 2
5	122 Payam Kaufling	\$7,900	0	Solo Sueldo	Stock Clerk	01 de 01 del 2007	31 de 12 del 2007	364 / 12 / 1
6	176 Jonathon Taylor	\$8,600	0,2	Sueldo + Comision	Sales Manager	01 de 01 del 2007	31 de 12 del 2007	364 / 12 / 1
7	176 Jonathon Taylor	\$8,600	0,2	Sueldo + Comision	Sales Representative	24 de 03 del 2006	31 de 12 del 2006	282 / 9 / 1
8	200 Jennifer Whalen	\$4,400	0	Solo Sueldo	Public Accountant	01 de 07 del 2002	31 de 12 del 2006	1644 / 54 / 5
9	200 Jennifer Whalen	\$4,400	0	Solo Sueldo	Administration Assistant	17 de 09 del 1995	17 de 06 del 2001	2100 / 69 / 6
10	201 Michael Hartstein	\$13,000	0	Solo Sueldo	Marketing Representative	17 de 02 del 2004	19 de 12 del 2007	1401 / 46 / 4

Acción	Resultado esperado
Verificar datos a utilizar	<pre>SELECT * FROM EMPLOYEES; SELECT * FROM JOBS; SELECT * FROM JOB_HISTORY;</pre>
Armar el primer paso	<pre>SELECT E.EMPLOYEE_ID "ID EMPLEADO", E.FIRST_NAME  ' '  E.LAST_NAME EMPLEADO, E.SALARY SUELDO, JH.START_DATE "FECHA DE CONTRATACION", JH.END_DATE "FECHA DE TERMINO" FROM EMPLOYEES E RIGHT JOIN JOB_HISTORY JH ON JH.EMPLOYEE_ID = E.EMPLOYEE_ID;</pre>
Agregar los datos faltantes	<pre>SELECT E.EMPLOYEE_ID "ID EMPLEADO", E.FIRST_NAME  ' '  E.LAST_NAME EMPLEADO, E.SALARY SUELDO, NVL(E.COMMISSION_PCT,0) COMISION, NVL2(E.COMMISSION_PCT,'Sueldo + Comision','Solo Sueldo') "COMISION DESCRIPCION", J.JOB_TITLE CARGO, JH.START_DATE "FECHA DE CONTRATACION", JH.END_DATE "FECHA DE TERMINO", ROUND((JH.END_DATE-JH.START_DATE)  ' / '   ROUND(MONTHS_BETWEEN(JH.END_DATE,JH.START_DATE))   ' / '   ROUND((JH.END_DATE-JH.START_DATE)/365) "DIAS / MESES / AÑOS CONTRATADOS" FROM EMPLOYEES E RIGHT JOIN JOB_HISTORY JH ON JH.EMPLOYEE_ID = E.EMPLOYEE_ID JOIN JOBS J ON J.JOB_ID=JH.JOB_ID;</pre>



Acción	Resultado esperado
Revisar los últimos detalles	<pre>SELECT E.EMPLOYEE_ID "ID EMPLEADO", E.FIRST_NAME  ' '  E.LAST_NAME EMPLEADO, TO_CHAR(E.SALARY,'\$99,999') SUELDO, NVL(E.COMMISSION_PCT,0) COMISION, NVL2(E.COMMISSION_PCT,'Sueldo + Comision','Solo Sueldo') "COMISION DESCRIPCION", J.JOB_TITLE CARGO, TO_CHAR (JH.START_DATE, 'dd "de" mm "del" yyyy') "FECHA DE CONTRATACION", TO_CHAR (JH.END_DATE, 'dd "de" mm "del" yyyy') "FECHA DE TERMINO", ROUND(MONTHS_BETWEEN(JH.END_DATE-JH.START_DATE)  ' / '   ROUND(MONTHS_BETWEEN(JH.END_DATE, JH.START_DATE))   ' / '   ROUND((JH.END_DATE-JH.START_DATE)/365) "DÍAS / MESES / AÑOS CONTRATADOS" FROM EMPLOYEES E RIGHT JOIN JOB_HISTORY JH ON JH.EMPLOYEE_ID = E.EMPLOYEE_ID JOIN JOBS J ON J.JOB_ID=JH.JOB_ID;</pre>

### Práctica guiada

Para guiar el proceso de incorporar nuevos conocimientos al proceso de consulta de datos, se sugiere que el docente guíe a los estudiantes para trabajar con JOIN y GROUP BY:

Acción	Resultado esperado																																																																		
Mostrar en qué departamentos trabajan los empleados y se muestre organizado por la dirección del departamento	<pre>SELECT E.EMPLOYEE_ID, E.FIRST_NAME  ' '  E.LAST_NAME NAME, D.DEPARTMENT_NAME, L.CITY, L.STREET_ADDRESS, C.COUNTRY_NAME FROM EMPLOYEES E JOIN DEPARTMENTS D ON D.DEPARTMENT_ID = E.DEPARTMENT_ID JOIN LOCATIONS L ON L.LOCATION_ID = D.LOCATION_ID JOIN COUNTRIES C ON C.COUNTRY_ID = L.COUNTRY_ID ORDER BY L.STREET_ADDRESS;</pre>																																																																		
Resultado esperado	<table border="1"> <thead> <tr> <th>EMPLOYEE_ID</th> <th>NAME</th> <th>DEPARTMENT_NAME</th> <th>CITY</th> <th>STREET_ADDRESS</th> <th>COUNTRY_NAME</th> </tr> </thead> <tbody> <tr><td>1</td><td>145 John Russell</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> <tr><td>2</td><td>146 Karen Partners</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> <tr><td>3</td><td>179 Charles Johnson</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> <tr><td>4</td><td>177 Jack Livingston</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> <tr><td>5</td><td>147 Alberto Errazuriz</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> <tr><td>6</td><td>148 Gerald Cambrault</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> <tr><td>7</td><td>149 Eleni Zlotkey</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> <tr><td>8</td><td>150 Peter Tucker</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> <tr><td>9</td><td>151 David Bernstein</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> <tr><td>10</td><td>152 Peter Hall</td><td>Sales</td><td>Oxford</td><td>Magdalen Centre, The Oxford Science Park United Kingdom</td><td></td></tr> </tbody> </table>	EMPLOYEE_ID	NAME	DEPARTMENT_NAME	CITY	STREET_ADDRESS	COUNTRY_NAME	1	145 John Russell	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom		2	146 Karen Partners	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom		3	179 Charles Johnson	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom		4	177 Jack Livingston	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom		5	147 Alberto Errazuriz	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom		6	148 Gerald Cambrault	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom		7	149 Eleni Zlotkey	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom		8	150 Peter Tucker	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom		9	151 David Bernstein	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom		10	152 Peter Hall	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom	
EMPLOYEE_ID	NAME	DEPARTMENT_NAME	CITY	STREET_ADDRESS	COUNTRY_NAME																																																														
1	145 John Russell	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
2	146 Karen Partners	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
3	179 Charles Johnson	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
4	177 Jack Livingston	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
5	147 Alberto Errazuriz	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
6	148 Gerald Cambrault	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
7	149 Eleni Zlotkey	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
8	150 Peter Tucker	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
9	151 David Bernstein	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
10	152 Peter Hall	Sales	Oxford	Magdalen Centre, The Oxford Science Park United Kingdom																																																															
Mostrar cuántos empleados tiene cada departamento	<pre>SELECT E.DEPARTMENT_ID, D.DEPARTMENT_NAME, COUNT (*) FROM EMPLOYEES E JOIN DEPARTMENTS D ON D.DEPARTMENT_ID = E.DEPARTMENT_ID GROUP BY E.DEPARTMENT_ID, D.DEPARTMENT_NAME ORDER BY E.DEPARTMENT_ID;</pre>																																																																		

Acción	Resultado esperado																																				
Resultado esperado	<table border="1"> <thead> <tr> <th>DEPARTMENT_ID</th> <th>DEPARTMENT_NAME</th> <th>COUNT(*)</th> </tr> </thead> <tbody> <tr><td>1</td><td>10 Administration</td><td>1</td></tr> <tr><td>2</td><td>20 Marketing</td><td>2</td></tr> <tr><td>3</td><td>30 Purchasing</td><td>6</td></tr> <tr><td>4</td><td>40 Human Resources</td><td>1</td></tr> <tr><td>5</td><td>50 Shipping</td><td>45</td></tr> <tr><td>6</td><td>60 IT</td><td>5</td></tr> <tr><td>7</td><td>70 Public Relations</td><td>1</td></tr> <tr><td>8</td><td>80 Sales</td><td>34</td></tr> <tr><td>9</td><td>90 Executive</td><td>3</td></tr> <tr><td>10</td><td>100 Finance</td><td>6</td></tr> <tr><td>11</td><td>110 Accounting</td><td>2</td></tr> </tbody> </table>	DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)	1	10 Administration	1	2	20 Marketing	2	3	30 Purchasing	6	4	40 Human Resources	1	5	50 Shipping	45	6	60 IT	5	7	70 Public Relations	1	8	80 Sales	34	9	90 Executive	3	10	100 Finance	6	11	110 Accounting	2
DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)																																			
1	10 Administration	1																																			
2	20 Marketing	2																																			
3	30 Purchasing	6																																			
4	40 Human Resources	1																																			
5	50 Shipping	45																																			
6	60 IT	5																																			
7	70 Public Relations	1																																			
8	80 Sales	34																																			
9	90 Executive	3																																			
10	100 Finance	6																																			
11	110 Accounting	2																																			

### Práctica independiente

Se recomienda un trabajo individual en el que el estudiante trabaje con ejercicios (máximo 5) donde aplique las herramientas aprendidas en los modelos, pero esta vez con el desafío de incorporar JOIN en las bases de datos creadas por ellos mismos, considerando situaciones cotidianas en las que el uso de las bases de datos es un facilitador de tareas en diferentes ámbitos de actividad humana, como el transporte, la salud o las comunicaciones. Para guiar a los estudiantes, se puede incorporar la resolución de problemas de manera algorítmica a partir de bases de datos.

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Consulta de base de datos	Genera Script con errores.	Genera Script de manera correcta, pero no se visualizan algunos datos.	Genera Script de manera correcta, visualizando todos los datos.
Herramienta JOIN	Usa la herramienta JOIN con errores.	Usa la herramienta JOIN, pero de manera ineficiente.	Usa la herramienta JOIN de manera comprensible y eficiente.

### Orientaciones al docente

**Para unificar conceptos disciplinares:** Para poder fortalecer los conocimientos de los estudiantes, lo primordial es la práctica mediante ejercicios, para ello se recomienda el uso de nuevas bases de datos y no solo utilizar la anterior mencionada “HR”, pues las consultas de base de datos nos sirven para poder conectar diferentes elementos que nos proporciona la base de datos; estos son la comunicación entre tablas para mantener la información lo más fiel que se pueda.

El concepto inicial del JOIN es la comunicación. Los datos guardados en una tabla a veces pueden ser códigos que tengan su significado en otra tabla. El usuario preferirá ver la información real antes de un código y tener que ir a revisar qué significa ese código como tal. Para eso sirve primordialmente esta herramienta.

Una cláusula JOIN se utiliza para combinar filas de dos o más tablas, según una columna relacionada entre ellas.

**Actitudes:** Para apoyar el desarrollo de las actividades, se recomienda el uso de computadores, ya que los trabajos que se hacen requieren de una conexión directa a una base de datos con información con la cual trabajar. En cuanto al desarrollo de la actividad individual, es necesario el computador para que el estudiante tenga autonomía y confianza a la hora de querer resolver problemas.

**Orientaciones para organizar e implementar el trabajo individual:** Se sugieren las siguientes indicaciones para guiar al estudiante en el trabajo individual:

- Elaborar un ejercicio amplio, como el de la construcción de conocimiento, o implementar varios ejercicios con dificultad moderada en los que se les solicite entre uno a dos cambios.
- Se puede trabajar con una base de datos ya hecha con anterioridad o una nueva con máximo 5 tablas.
- Determinar un tiempo específico para poder realizar las actividades.
- Dar retroalimentación del ejercicio.

## Actividad de desempeño 2

### Propósito

En esta actividad, se trabajará el cómo algunas “consultas” no se pueden obtener con un solo intento siendo necesario agrupar un número de datos para lograr conseguir el resultado esperado al implementar un algoritmo. En la medida que incorporen este desempeño, los estudiantes podrán comprender que, al momento de hacer una consulta de base de datos, se requiere tener claridad en la especificidad de la información que realmente se necesita conseguir para aplicar soluciones algorítmicas a diferentes problemáticas.

### Objetivos de Aprendizaje

**OA4.** Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada.

**(Generalizar y Transferir)**

**OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista.

**(Generalizar y Transferir)**

### Conocimiento esencial

Representación y análisis de datos.

### Tiempo estimado

3 horas pedagógicas.



- Se necesita saber los empleados que tienen un salario menor al promedio general de los empleados:

	EMPLEADO	SALARIO
1	132	\$2,100
2	128	\$2,200
3	136	\$2,200
4	135	\$2,400
5	127	\$2,400
6	182	\$2,500
7	191	\$2,500

Acción	Resultado esperado	Preguntas de apoyo
Ejercicio 1	<pre>SELECT EMPLOYEE_ID "IDENTIFICACION EMPLEADO", FIRST_NAME  ' '  LAST_NAME EMPLEADO, JOB_ID,TO_CHAR(SALARY,'\$99,999') SALARIO FROM EMPLOYEES WHERE SALARY = (SELECT SALARY FROM EMPLOYEES WHERE EMPLOYEE_ID = 204);</pre>	
Ejercicio 2	<pre>SELECT AVG(E.ID1) "PROMEDIO DE EMPLEADOS POR DEPARTAMENTO" FROM (SELECT COUNT(*) AS ID1 FROM EMPLOYEES R GROUP BY R.DEPARTMENT_ID) E;</pre>	¿Cómo hacer el promedio de un contador?
Ejercicio 3	<pre>SELECT EMPLOYEE_ID EMPLEADO, TO_CHAR(SALARY,'\$99,999') SALARIO FROM EMPLOYEES WHERE SALARY &lt; (SELECT AVG(SALARY) FROM EMPLOYEES) ORDER BY SALARY;</pre>	

### Práctica guiada

Para implementar algoritmos que permitan trabajar con datos, se propone que el docente guíe a los estudiantes mostrándoles nuevos conceptos en cuanto a consultas de base de datos, en los cuales se trabajará con JOIN, GROUP BY y SUB-CONSULTAS:

Acción	Resultado esperado																																				
Conseguir los departamentos en donde no hay empleados	<pre>SELECT DEPARTMENT_ID "ID DEPARTAMENTO",DEPARTMENT_NAME "NOMBRE DEPARTAMENTO" FROM DEPARTMENTS WHERE DEPARTMENT_ID &lt;&gt; ALL (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT_ID IS NOT NULL GROUP BY DEPARTMENT_ID) ORDER BY DEPARTMENT_ID;</pre>																																				
Resultado esperado	<table border="1"> <thead> <tr> <th>ID DEPARTAMENTO</th> <th>NOMBRE DEPARTAMENTO</th> <th>ID DEPARTAMENTO</th> <th>NOMBRE DEPARTAMENTO</th> </tr> </thead> <tbody> <tr><td>1</td><td>120 Treasury</td><td>9</td><td>200 Operations</td></tr> <tr><td>2</td><td>130 Corporate Tax</td><td>10</td><td>210 IT Support</td></tr> <tr><td>3</td><td>140 Control And Credit</td><td>11</td><td>220 NOC</td></tr> <tr><td>4</td><td>150 Shareholder Services</td><td>12</td><td>230 IT Helpdesk</td></tr> <tr><td>5</td><td>160 Benefits</td><td>13</td><td>240 Government Sales</td></tr> <tr><td>6</td><td>170 Manufacturing</td><td>14</td><td>250 Retail Sales</td></tr> <tr><td>7</td><td>180 Construction</td><td>15</td><td>260 Recruiting</td></tr> <tr><td>8</td><td>190 Contracting</td><td>16</td><td>270 Payroll</td></tr> </tbody> </table>	ID DEPARTAMENTO	NOMBRE DEPARTAMENTO	ID DEPARTAMENTO	NOMBRE DEPARTAMENTO	1	120 Treasury	9	200 Operations	2	130 Corporate Tax	10	210 IT Support	3	140 Control And Credit	11	220 NOC	4	150 Shareholder Services	12	230 IT Helpdesk	5	160 Benefits	13	240 Government Sales	6	170 Manufacturing	14	250 Retail Sales	7	180 Construction	15	260 Recruiting	8	190 Contracting	16	270 Payroll
ID DEPARTAMENTO	NOMBRE DEPARTAMENTO	ID DEPARTAMENTO	NOMBRE DEPARTAMENTO																																		
1	120 Treasury	9	200 Operations																																		
2	130 Corporate Tax	10	210 IT Support																																		
3	140 Control And Credit	11	220 NOC																																		
4	150 Shareholder Services	12	230 IT Helpdesk																																		
5	160 Benefits	13	240 Government Sales																																		
6	170 Manufacturing	14	250 Retail Sales																																		
7	180 Construction	15	260 Recruiting																																		
8	190 Contracting	16	270 Payroll																																		
Conseguir el departamento con el mayor número de empleados	<pre>SELECT D.DEPARTMENT_NAME DEPARTAMENTO,COUNT(*) " TOTAL DE EMPLEADOS" FROM EMPLOYEES E JOIN DEPARTMENTS D ON E.DEPARTMENT_ID=D.DEPARTMENT_ID GROUP BY D.DEPARTMENT_NAME HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM EMPLOYEES GROUP BY DEPARTMENT_ID);</pre>																																				
Resultado esperado	<table border="1"> <thead> <tr> <th>DEPARTAMENTO</th> <th>TOTAL DE EMPLEADOS</th> </tr> </thead> <tbody> <tr> <td>1 Shipping</td> <td>45</td> </tr> </tbody> </table>	DEPARTAMENTO	TOTAL DE EMPLEADOS	1 Shipping	45																																
DEPARTAMENTO	TOTAL DE EMPLEADOS																																				
1 Shipping	45																																				

### Práctica independiente

Se recomienda un trabajo individual en donde el estudiante trabaje con ejercicios (máximo 5), en donde aplique las herramientas aprendidas en los modelos, pero esta vez con el desafío de incorporar GROUP BY y SUB-CONSULTAS en bases de datos creadas por ellos mismo, considerando situaciones cotidianas, donde el uso de las bases de datos es un facilitador de tareas en diferentes ámbitos de la actividad humana, como el transporte, la salud o las comunicaciones. Para guiar a los estudiantes, se puede incorporar la resolución de problemas de manera algorítmica a partir de bases de datos.

## Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Consulta de base de datos	Genera Script con errores.	Genera Script de manera correcta, pero no se visualizan algunos datos.	Genera Script de manera correcta visualizando todos los datos.
Herramientas GROUP BY / SUBQUERIES	Usa las herramientas GROUP BY/SUBQUERIES con errores.	Usa las herramientas GROUP BY/SUBQUERIES, pero de manera ineficiente.	Usa las herramientas GROUP BY/SUBQUERIES de manera comprensible y eficiente.

## Orientaciones al docente

**Para unificar conceptos disciplinares:** Trabajar con dos consultas. Debe ser algo confuso en un inicio y puede provocar ciertos enredos al momento de comprender y planear lo que se debe hacer para resolver la problemática a nivel inicial.

Términos del GROUP BY:

- La declaración **GROUP BY** agrupa las filas que tienen los mismos valores en filas de resumen, como "encontrar el número de clientes en cada país".
- La instrucción **GROUP BY** se utiliza a menudo con funciones de agregado (COUNT(), MAX(), MIN(), SUM(), AVG()) a grupa el conjunto de resultados de una o más columnas.

Términos de las SUB-CONSULTAS:

- Una **subconsulta** o consulta interna o una consulta anidada es una consulta dentro de otra consulta SQL e incrustada dentro de la cláusula WHERE.
- Una **subconsulta** se utiliza para devolver datos que se utilizarán en la consulta principal como condición para restringir aún más los datos que se recuperarán.
- Las **subconsultas** se pueden usar con las instrucciones SELECT, INSERT, UPDATE y DELETE junto con los operadores como =, <, >, >=, <=, IN, BETWEEN, etc.

Hay algunas reglas que las subconsultas deben seguir:

- Las subconsultas deben ir entre paréntesis.
- Una subconsulta solo puede tener una columna en la cláusula SELECT, a menos que haya varias columnas en la consulta principal para que la subconsulta compare sus columnas seleccionadas.
- Un comando ORDER BY no se puede usar en una subconsulta, aunque la consulta principal puede usar un ORDER BY. El comando GROUP BY se puede utilizar para realizar la misma función que ORDER BY en una subconsulta.
- Las subconsultas que devuelven más de una fila solo se pueden usar con operadores de valor múltiple, como el operador IN.



- *La lista SELECT no puede incluir referencias a valores que se evalúen como BLOB, ARRAY, CLOB o NCLOB.*
- *Una subconsulta no puede incluirse inmediatamente en una función establecida.*
- *El operador BETWEEN no se puede utilizar con una subconsulta. Sin embargo, el operador BETWEEN se puede utilizar dentro de la subconsulta.*

**Actitudes:** Para apoyar el desarrollo de las actividades se recomienda el uso de computadores, ya que los trabajos que se hacen requieren de una conexión directa a una base de datos con información con la cual trabajar. En cuanto al desarrollo de la actividad individual, es necesaria esa herramienta para que el estudiante tenga autonomía y confianza a la hora de querer resolver problemas.

**Orientaciones para organizar e implementar el trabajo individual:** Se sugieren las siguientes indicaciones para guiar al estudiante en el trabajo individual.

- Elaborar un ejercicio amplio como en la construcción de conocimiento o implementar varios ejercicios con dificultad moderada en el cual se le solicite dos consultas.
- Trabajar con una base de datos nueva con máximo 10 tablas.
- Dar retroalimentación del ejercicio.

## Actividad de desempeño 3

### Propósito

Esta actividad está enfocada en el conocimiento y uso del lenguaje de programación Python, aplicando un proceso lógico que vincule la implementación de un algoritmo con dicho lenguaje. Al mismo tiempo, se propicia la elaboración de soluciones a situaciones problemática mediante la elaboración de algoritmos y la programación, visualizando las diferencias en cuanto a hacer un pseudocódigo y un código.

### Objetivos de Aprendizaje

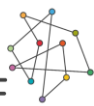
**OA6.** Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

### Conocimiento esencial

- Algoritmos y programación: lenguajes, métodos y entornos de programación.
- Representación y análisis de datos.

### Tiempo estimado

3 horas pedagógicas.



## Desarrollo de la actividad

### Situación experiencial

Se sugiere al docente que presente un pseudocódigo y un código, y hacer notar a los estudiantes que en ambos se utiliza el mismo algoritmo, el cual resuelve detectar si un número ingresado es par o impar ante, por ejemplo, la necesidad de configurar una máquina que detecte la apenadura de llantas de un auto:

<pre><b>Escribir</b> "Escriba un número: " <b>Leer</b> Num  <b>Si</b> Num % 2 = 0 <b>Entonces</b> ..... <b>Escribir</b> Num, " es un numero par" <b>SiNo</b> ..... <b>Escribir</b> Num, " es un numero impar" <b>Fin Si</b></pre>	<pre>print("Ingrese Número : ") num = int(input()) cociente = round(num / 2) residuo = cociente * 2 resultado = num - residuo if resultado == 0:     print("NÚMERO PAR") else:     print("NÚMERO IMPAR")</pre>
---	--

Los estudiantes tendrán que analizar los algoritmos, establecer las diferencias entre ellos y comentar guiados por las siguientes preguntas.

- ¿Cómo están estructurados los algoritmos?
- ¿Qué diferencias hay entre un pseudocódigo y un código?
- ¿Hay alguna semejanza en cuanto a funcionalidad?

### Construcción de conocimiento

Para construir el conocimiento utilizando las bases del algoritmo y crear pseudocódigos, se sugiere al docente que indique a los estudiantes recordar las medidas aprendidas con anterioridad de una manera que funcione para un código en un programa de programación profesional. Como ejercicio de práctica, es necesario que los estudiantes puedan elaborar las nuevas sintaxis de la forma más realista posible, de tal forma que sea posible verlas en el nuevo lenguaje de programación, como Python. Para esto, se trabajará con problemas ya antes vistos en PSeint, de forma que el proceso de transferencia tenga mayor significado.

Como ejercicio, se realizará una calculadora con las funciones de “sumar, restar, multiplicar y dividir” con solo dos valores a elegir para realizar el cálculo.

Acción	Resultado esperado
Definir los valores a utilizar	<pre>opc = int valor1 = int valor2 = int total = float</pre>
Crear menú e ingresar valores a calcular	<pre>print("Calculadora") print("""1: Sumar 2: Restar 3: Multiplicar 4: Dividir""") opc = int(input("Ingrese opcion: "))  valor1 = int(input("Ingrese valor1: ")) valor2 = int(input("Ingrese valor2: "))</pre>
Hacer el cálculo dependiendo de la opción que ingreso	<pre>if opc == 1:     total = valor1 + valor2     print("La suma de ",valor1," + ",valor2," = ",total) elif opc == 2:     total = valor1 - valor2     print("La resta de ",valor1," - ",valor2," = ",total) elif opc == 3:     total = valor1 * valor2     print("La multiplicacion de ",valor1," * ",valor2," = ",total) elif opc == 4:     total = valor1 / valor2     print("La division de ",valor1," / ",valor2," = ",total)</pre>

### Práctica guiada

Para guiar a los estudiantes, se sugiere un ejercicio de la misma categoría, pero con un concepto más real que se podría ver reflejado en la cotidianidad. Por ejemplo, se necesita determinar el cálculo del impuesto en la renta de unos departamentos en alguna parte del país. Los datos a seguir son los siguientes:

CATEGORIA	RENTA	IMPUESTO
Bronce	Menos de \$1 440 000	3%
Plata	Entre \$1 440 000 y \$6 000 000	5%
Oro	Entre \$6 000 000 y \$11 400 000	10%
Diamante	Mas de \$11 400 000	20%

Sobre este contexto, se sugiere como desafío a los estudiantes: escribir un programa que pregunte al usuario su renta anual y muestre en el terminal lo que debe pagar de impuesto en total:

Acción	Resultado esperado
<p>Programación Python esperada para el desarrollo del ejercicio</p>	<pre> 1 # Preguntar al usuario por la renta 2 renta = float(input("¿Cuál es tu renta anual? ")) 3 # Condicional para determinar el tipo impuesto 4 if renta &lt; 1440000: 5     tipo = 3 6 elif renta &gt;= 1440000 and renta &lt; 6000000: 7     tipo = 5 8 elif renta &gt;= 6000000 and renta &lt; 11400000: 9     tipo = 10 10 elif renta &gt;= 11400000: 11     tipo = 20 12 13 # Mostrar por pantalla el impuesto a pagar 14 print("El impuesto a pagar es: \$", renta * tipo / 100, sep='') </pre>

### Práctica independiente

Se sugiere que los estudiantes realicen un trabajo individual con al menos 5 ejercicios con condicionales, para así poder usar el nuevo lenguaje de programación aplicando un proceso lógico; por ejemplo, considerando la siguiente situación:

Una pizzería llamada Luigi's ofrece pizzas vegetarianas y tradicionales a sus clientes, los ingredientes para cada tipo de pizza son los siguientes:

- Vegetariano: choclo, pimentón y tofu.
- Tradicional: jamón, pepperoni y salchichas.

Se debe escribir un programa que pregunte al usuario si quiere una pizza vegetariana o no, y en función de su respuesta le mostrara un menú en el cual elegir los ingredientes a elegir por categoría. Solo se puede elegir un ingrediente además de la mozzarella y el tomate que están en todas las pizzas. Al final, debe mostrar en pantalla los detalles de la pizza elegida por el cliente.

## Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Código	Elabora un código, con errores.	Elabora un código con algunos errores.	Elabora un código sin errores.
Sintaxis	Usa la sintaxis con errores.	Usa la sintaxis con algunos errores.	Usa la sintaxis sin errores.

## Orientaciones al docente

**Para unificar conceptos disciplinares:** el algoritmo es la base principal de todo entendimiento lógico a nivel computacional en el lenguaje de la programación. La aplicación utilizada para desarrollar los códigos de los ejercicios es Microsoft Studio Code, utilizando el lenguaje de programación “Python”, a pesar de ser un lenguaje estricto en cuanto a escritura se refiere, y es uno de los más sencillos para aprender lo que es codificación profesional:

*En PSeint fue posible ver que en cuanto a condicionales existen el Si y el Según, en este lenguaje solo existe la derivación del Si (If, elif, else).*

*“Microsoft Studio Code” es un editor de texto que a su vez facilita la implementación de sugerencias a la hora de codificar en el lenguaje impuesto, también se puede utilizar “Sublime Text” como medio para realizar los ejercicios.*

**Actitudes:** Para apoyar el desarrollo de las actividades, se recomienda utilizar en todo momento computadores configurados para la ocasión con sus respectivos programas y controladores para poder trabajar en los problemas que vayan surgiendo a lo largo del conocimiento de este lenguaje. En cuanto al desarrollo de la actividad individual, esta se hará de esta manera para que el estudiante sienta autonomía y utilice sus conocimientos previos en cuanto al funcionamiento del algoritmo y así pueda resolver los problemas.

**Orientaciones para organizar e implementar el trabajo individual:** Se sugieren las siguientes indicaciones para guiar al estudiante en el trabajo individual:

- Elaborar cinco ejercicios como en la construcción de conocimiento.
- Determinar un tiempo específico para poder realizar las actividades.
- Evaluar el impacto de su resultado a nivel social o ético de acuerdo con el problema que resuelve.
- Dar retroalimentación del ejercicio.

## Actividad de desempeño 4

### Propósito

Esta actividad desafía al estudiante a aprender cómo no repetir código y cómo poder trabajar las herramientas de los bucles en el lenguaje de programación Python. La idea es que luego puedan elaborar e implementar algoritmos usando un nuevo lenguaje de programación, generalizando y transfiriendo los aprendizajes asociados a la programación de código y pseudocódigo a realidades o necesidades cotidianas propias o de otros.

### Objetivos de Aprendizaje

**OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista.

**(Generalizar y Transferir)**

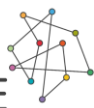
**OA6.** Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

### Conocimiento esencial

- Algoritmos y programación: lenguajes, métodos y entornos de programación.
- Representación y análisis de datos.
- Impacto social de la computación: interacciones, cultura, seguridad y ética.
- Lenguaje de programación: Python
- Sintaxis básicas (tipos, operadores, variables, funciones, listas, tuplas, diccionarios)
- Condicionales (if y else)

### Tiempo estimado

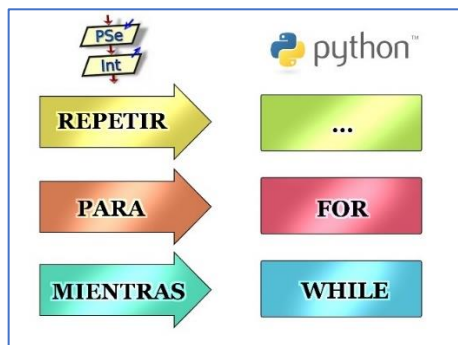
3 horas pedagógicas.



## Desarrollo de la actividad

### Situación experiencial

Se sugiere al docente presentar a los estudiantes una imagen comparativa con respecto a los bucles entre PSeint y Python. Se sugiere plantear un contexto donde se requiera un menú para albergar dos opciones y que una de esas opciones sea calcular el promedio de personas en un restaurant dependiendo de una cantidad seleccionable de personas a disposición del usuario. Por ejemplo, calcular el promedio de personas que pueden ingresar dado el aforo de lugar:



```
while opc == 1 or opc == 2:
```

```
Mientras opc = 1 o opc = 2 Hacer
```

```
for cont in range(0,n):
```

```
Para cont←1 Hasta Num_Notas Hacer
```

Luego, los estudiantes tendrán que analizar y comentar los bucles en base a las siguientes preguntas:

- ¿Por qué no existe el repetir en el lenguaje de programación Python?
- ¿Cuáles son las diferencias fundamentales entre Python y PSeint?

### Construcción de conocimiento

Los algoritmos son una pieza fundamental en cuanto a establecer un orden en los códigos al momento de programar, por lo que para la siguiente actividad se sugiere trabajar con los clásicos menús que se trabajaron en módulos anteriores con PSeint.

Con respecto a lo anterior se trabajará con un menú donde habrá dos opciones a elegir y son las siguientes:

- Calcular el promedio dependiendo de la cantidad de notas que ingrese.



- Una tabla de multiplicar.

Acción	Resultado esperado
Elaborar el menú con las dos opciones previamente indicadas	<pre> 1  opc = 1 2  while opc == 1 or opc == 2: 3      opc = int(input("""INGRESE UNA OPCION 4          1- Calcular promedio 5          2- Tabla de multiplicar 6          3- Salir 7          """)) </pre>
Elaborar la primera opción, ingresando la cantidad de notas a ingresar, entrando a un bucle "for" para así no repetir código y repetir la cantidad previamente ingresada por el usuario para finalizar con el promedio final calculado.	<pre> 9      if opc == 1: 10         suma = 0 11         n = int(input("INGRESE LA CANTIDAD DE NOTAS : ")) 12         for cont in range(0,n): 13             nota = int(input("NOTA : ")) 14             suma = suma + nota 15         print("PROMEDIO : {:.2f}".format(suma/n)) </pre>
Elaborar la segunda opción, ingresando el número para el que se harán sus respectivas multiplicaciones.	<pre> 17         if opc == 2: 18             print("Ingrese un Número: ") 19             num = int(input()) 20             for cont in range(1,13): 21                 print(cont, " X ", num, " = ", (cont*num)) </pre>

### Práctica guiada

Siguiendo el ejercicio anterior, se sugiere que el docente sitúe a los estudiantes en un contexto real cercano o cotidiano, por ejemplo:

Diagnóstico	IMC
PESO INSUFICIENTE	<18.5
PESO NORMAL	18.5 – 12.9
SOBREPESO	25 – 29.9
OBESIDAD GRADO 1	30 – 34.9
OBESIDAD GRADO 2	35 – 39.9
OBESIDAD GRADO 3	≥ 40

Un hospital para el área de nutrición requiere de un programa que permita realizar un cálculo de índice de masa corporal. Para esto se requiere la información de su peso (kg) y altura (metros).

La fórmula para sacar el cálculo es el siguiente:

$$IMC = \frac{\text{Peso (kg)}}{\text{Altura}^2 \text{ (m)}}$$

Acción	Resultado esperado
<p>Iniciar las variables, dar a elegir cuántos clientes se harán el diagnóstico del IMC.</p> <p>Se hace el cálculo con la fórmula y se verifica en qué categoría se encuentra el cliente.</p>	<pre> 30  ipe = 0 31  pno = 0 32  sg = 0 33  ob1 = 0 34  ob2 = 0 35  ob3 = 0 36 37  print("Bienvenido a IMC") 38  cant_cli = int(input("Indique la cantidad de clientes: ")) 39 40  for i in range(0,cant_cli): 41      print("-----") 42      print("Ingrese peso del cliente N°",i+1) 43      peso = int(input()) 44      print("Ingrese altura en metros del cliente N°",i+1) 45      talla = int(input()) 46      print("El cliente ingresado esta en la categoria:") 47      imc = peso / talla ** 2 48 49      if imc &lt; 18.5: 50          print("Peso Insuficiente") 51          ipe = ipe + 1 52      elif imc &gt;= 18.5 and imc &lt;= 24.9: 53          print("Peso Normal") 54          pno = pno + 1 55      elif imc &gt;= 25 and imc &lt;= 29.9: 56          print("SobrePeso") 57          sg = sg + 1 58      elif imc &gt;= 30 and imc &lt;= 34.9: 59          print("Obesidad Grado I") 60          ob1 = ob1 + 1 61      elif imc &gt;= 35 and imc &lt;= 39.9: 62          print("Obesidad Grado II") 63          ob2 = ob2 + 1 64      elif imc &gt;= 40 : 65          print("Obseidad Grado III") 66          ob3 = ob3 + 1 67      print("-----") 68 69  print("-----") 70  print("Cantidad de clientes por categoria:") 71  cant_cliente(ipe,pno,sg,ob1,ob2,ob3) 72  print("-----") 73  print("La cantidad con mas clientes es:") 74  cant_imc(ipe,pno,sg,ob1,ob2,ob3) 75  print("-----") </pre>
<p>Elaborar una función para verificar la cantidad de clientes por categoría.</p>	<pre> 1  def cant_cliente(ipe,pno,sg,ob1,ob2,ob3): 2      if ipe &gt; 0: 3          print("Cantidad de Clientes con Insuficiente Peso: ",ipe) 4      if pno &gt; 0: 5          print("Cantidad de Clientes con Peso Normal: ",pno) 6      if sg &gt; 0: 7          print("Cantidad de Clientes con SobrePeso Grado I: ",sg) 8      if ob1 &gt; 0: 9          print("Cantidad de Clientes con Obesidad Grado I: ",ob1) 10     if ob2 &gt; 0: 11         print("Cantidad de Clientes con Obesidad Grado II: ",ob2) 12     if ob3 &gt; 0: 13         print("Cantidad de Clientes con Obesidad Grado III: ",ob3) </pre>
<p>Elaborar una segunda función para ver cuál categoría es la que tiene la mayor cantidad de clientes.</p>	<pre> 15  def cant_imc(ipe,pno,sg,ob1,ob2,ob3): 16      if ipe &gt;= pno and ipe &gt;= sg and ipe &gt;= ob1 and ipe &gt;= ob2 and ipe &gt;= ob3: 17          print(" Mas Clientes en Insuficiente Peso") 18      if pno &gt;= ipe and pno &gt;= sg and pno &gt;= ob1 and pno &gt;= ob2 and pno &gt;= ob3: 19          print("Mas Clientes en Peso Normal") 20      if sg &gt;= ipe and sg &gt;= pno and sg &gt;= ob1 and sg &gt;= ob2 and sg &gt;= ob3: 21          print("Mas Clientes en SobrePeso Grado I") 22      if ob1 &gt;= ipe and ob1 &gt;= pno and ob1 &gt;= sg and ob1 &gt;= ob2 and ob1 &gt;= ob3: 23          print("Mas Clientes en Obesidad Grado I") 24      if ob2 &gt;= ipe and ob2 &gt;= pno and ob2 &gt;= sg and ob2 &gt;= ob1 and ob2 &gt;= ob3: 25          print("Mas Clientes en Obesidad Grado II") 26      if ob3 &gt;= ipe and ob3 &gt;= pno and ob3 &gt;= sg and ob3 &gt;= ob1 and ob3 &gt;= ob2: 27          print("Mas Clientes en Obesidad Grado III") </pre>

### Práctica independiente

Se sugiere que los estudiantes realicen un trabajo individual con al menos 3 ejercicios con bucles para así poder usar el nuevo lenguaje de programación, por ejemplo: elaborar un programa que permita almacenar los nombres y promedios de la cantidad alumnos de un curso, las notas deben estar en un rango de 10 y 70. Mostrar el alumno con la mayor nota.

### Evaluación formativa

Para verificar aprendizajes, se sugiere considerar una rúbrica con los siguientes criterios:

Criterio	Inicial	Intermedio	Avanzado
Código	Elabora un código, con errores.	Elabora un código con algunos errores.	Elabora un código sin errores.
Sintaxis	Usa la sintaxis con errores.	Usa la sintaxis con algunos errores.	Usa la sintaxis sin errores.

### Orientaciones al docente

**Para unificar conceptos disciplinares:** Con el conocimiento obtenido para diseñar un algoritmo, todo se hace mucho más fácil al momento de implementarlo en un lenguaje de programación, pero eso no significa que los estudiantes deban dejar de aprender cosas nuevas con respecto al lenguaje de programación, ya que muchas cosas mínimas terminan cambiando a favor del lenguaje utilizado para trabajar.

Por estas razones en PSeint se tenía el *Repetir*, *Para* y *Mientras*, pero en el nuevo lenguaje se tiene las variantes del Para y Mientras (For y While).

**Actitudes:** Para apoyar el desarrollo de las actividades, se recomienda utilizar en todo momento computadores. En cuanto al desarrollo de la actividad individual, esta se hará de esta manera para que el estudiante sienta autonomía y utilice sus conocimientos previos en cuanto al funcionamiento del algoritmo y así pueda resolver los problemas.

**Orientaciones para organizar e implementar:** Se sugieren las siguientes indicaciones para guiar al estudiante en el trabajo individual.

- Elaborar tres ejercicios como en la construcción de conocimiento.
- Determinar un tiempo específico para poder realizar las actividades.
- Evaluar el impacto de su resultado a nivel social o ético de acuerdo con el problema que resuelve.
- Dar retroalimentación del ejercicio.

# MÓDULOS ELECTIVOS

## PENSAMIENTO COMPUTACIONAL

## Módulo Aprendizaje Basado en Proyectos

### Visión panorámica

#### Gran idea

El comportamiento de un sistema computacional (dispositivos, *hardware* y *software*) depende de la secuencia de instrucciones dadas por el usuario que automatizarán el desempeño de una tarea para ejecutar una acción o resolver un problema.

#### Objetivos de aprendizaje

**OA2.** Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

**OA3.** Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

#### Conocimientos esenciales

- Modelos entidad-relación y simulación de sistemas.
- Sistemas computacionales: dispositivos, *hardware* y *software*, y resolución de problemas.

#### Tiempo estimado

6 semanas (12 horas)

## Propósito

Este proyecto tiene como propósito la creación de una base de datos orientada a un diseño computacional profesional, incorporando la formación de datos de acuerdo con el problema que sea necesario solucionar.

Para iniciar, los estudiantes analizan los elementos esenciales del problema, para ello, en equipos de trabajo deberán distinguir la información específica que conforma la base de datos y sus respectivas relaciones. También, podrán contrastar con otros equipos, sus resultados y retroalimentar el proceso que siguieron.

Luego, a partir del análisis y obtención de la información del problema planteado, los estudiantes modelan la base de datos para poder verificar si sus datos tienen coherencia estructural para resolver el problema en cuanto a lo esencial del mismo.

Para finalizar, crean la base de datos en un contenedor (Oracle, SQL Server o SQL Lite) y trabajar en la creación de entidades para ella. Para esto, desarrollan una consulta que pueda crear las tablas e insertar los datos que puedan tener significado en el problema. Por último, exponen la base de datos creada y buscarán la posibilidad de concretar en la realidad sus ideas, dando contexto técnico y funcional a sus creaciones

### Problema central

En la actualidad, diferentes ámbitos de la actividad humana requieren manejar datos con la finalidad de que la información que se genera, por ejemplo, en actividades comerciales, se aprovechen de forma efectiva. En este sentido, ampliar el conocimiento de los jóvenes y adultos en el uso de datos a nivel de empresas, cobra valor en cuanto a desafiarlos a analizar e investigar posibles entornos laborales o de emprendimientos futuros, en donde se requiera el manejo de datos específicos para desempeñar tareas o resolver problemas.

Problema:

*Una empresa de alimentos en Chile requiere una base de datos centralizada donde almacenar todos sus productos, dependiendo del stock de cada tienda y sus trabajadores. Estos productos vienen principalmente de los proveedores en el que se conoce su información personal. Por otra parte, proveedores pueden estar o no registrados en la empresa, si están registrados se les solicita su RUT, nombre, dirección, sexo y opcionalmente el teléfono y correo electrónico.*

*La empresa tiene al menos una tienda en cada región de Chile. Estas tiendas se manejan en base a un identificador, nombre y su respectiva dirección descrita con la región y comuna. Cada tienda tiene su propia cantidad de stock de productos, los productos tienen un identificador, nombre, descripción, fecha de llegada, precio y categoría del producto. En la boleta debe ir el empleado que realizó la venta junto a su respectivo cliente, además de contener la fecha del día en el que se factura. La boleta debe tener el detalle de la compra, con los productos que se seleccionaron, su cantidad y el precio total, además debe indicar de que tienda se realizó la compra.*

Las categorías que tienen los productos alimenticios son las siguientes:

- *Grupo I: Leche y derivados. Son alimentos plásticos. En ellos predominan las proteínas.*
- *Grupo II: Carnes pescados y huevos. Alimentos plásticos. En ellos predominan las proteínas.*
- *Grupo III: Legumbres, frutos secos y patatas. Alimentos energéticos, plásticos y reguladores. En ellos predominan los glúcidos, pero también poseen cantidades importantes de proteínas, vitaminas y minerales.*
- *Grupo IV: Hortalizas. Alimentos reguladores. En ellos predominan las vitaminas y minerales.*
- *Grupo V: Frutas. Alimentos reguladores. En ellos predominan las vitaminas y minerales.*
- *Grupo VI: Cereales. Alimentos energéticos. En ellos predominan los glúcidos.*
- *Grupo VII: Mantecas y aceites. Alimentos energéticos. En ellos predominan los lípidos.*

Los trabajadores se identifican mediante su RUT, nombre, apellido, dirección, sexo y su cargo, asimismo, ellos pueden recibir muchos pedidos en el día.

Los cargos que tienen los trabajadores son los siguientes:

- *Gerente.*
- *Cajero.*
- *Reponedor.*
- *Bodeguero.*
- *Especialista de sección tales como frutería, carnicería o pescadería, panadería entre otros.*

## Propósito

Crear una base de datos centralizada funcional orientada a un diseño computacional profesional, que, resuelva un problema en un ámbito de la actividad humana que sea cerca o significativa para los estudiantes, modelando a su vez la base de datos y elaborando un informe descriptivo que dé cuenta del proceso.

## Objetivos de aprendizaje

**OA2.** Organizar datos de una situación problemática estableciendo secuencias e identificando fallos y defectos, trabajando con honestidad y considerando criterios de usabilidad, funcionamiento y función. **(Evaluar)**

**OA3.** Modelar datos de una situación problemática mediante representaciones, simulaciones simples, bloques de programación y diagramas, aprovechando las herramientas disponibles. **(Modelar)**

## Preguntas

- ¿Cuál es el propósito de la base de datos solicitada?
- ¿Cómo la base de datos ayuda con el manejo de datos críticos?
- ¿Cuáles son los elementos necesarios para diseñar una base de datos?
- ¿Qué medios permiten modelar la base de datos?
- ¿Cómo se puede identificar los datos correspondientes dentro de un problema?
- ¿Qué debo hacer cuando no logro encontrar las relaciones correspondientes en el problema?
- ¿Cómo se puede crear la base de datos centralizada para responder al problema?

## Tipo de proyecto

Proyecto intradisciplinario

## Producto

Base de datos centralizada, que resuelve un problema de control de stock de una empresa de distribución de alimentos a nivel nacional y un informe descriptivo del proceso de solución que permita generalizar y transferir la solución a otros contextos que requieran del control de datos.

## Habilidades y actitudes para el siglo XXI

### Habilidades:

Maneras de Pensar: Creatividad e Innovación.

Maneras de Trabajar: Comunicación y colaboración.



### Actitudes:

Pensar con perseverancia y proactividad para encontrar soluciones innovadoras a los problemas.

Trabajar con honestidad en la realización de tareas colaborativas con miras al logro de una meta común.

### Etapas

**Etapa 1:** En esta etapa los estudiantes descomponen y analizan el problema principal, aplicando el conocimiento de encontrar las tablas, columnas y relaciones. Asimismo, idean por medio del diseño el modelo de base de datos.

**Etapa 2:** En esta etapa los estudiantes modelarán una base de datos con el análisis que se hizo en la etapa 1. Con esto deberán ordenar y unir todo para que la base de datos cumpla con lo establecido en el problema.

**Etapa 3:** En esta etapa los estudiantes crean la base de datos. Esto dada la consulta de base de datos y a su vez insertar los datos que deben ser coherentes al problema que hay resolver.

**Etapa 4:** En esta etapa los estudiantes harán un informe descriptivo respecto del proceso de obtención de la solución, indicando inconvenientes y explicando el proceso creativo. Finalmente, prueban y comparten con sus pares el producto obtenido.

### Cronograma semanal

#### Etapa 1:

En esta etapa los estudiantes conforman equipos de trabajo, idealmente de tres integrantes, descomponen y analizan el problema principal, para que encontrar los datos necesarios para la creación y modelamiento de la base de datos:

- Leen el problema y lo descomponen para encontrar cada dato y analizan cada aspecto para identificar lo esencial del problema. Además, idean el modelo de base de datos que resolverá el problema, proponiendo una solución individual para luego decidir en equipo la mejor opción.
- A partir del análisis en equipos, identifican las entidades y relaciones que tiene el problema. Con estos datos se crean los primeros diseños que tendrá el modelo. Cada integrante debería realizar un diseño.
- Analizan los primeros diseños y verifican los datos para encontrar nuevos datos que puedan servir y complementar al problema.
- Explican detalladamente por qué se escogió el diseño final en equipo.

#### Etapa 2:

En esta etapa, los estudiantes desarrollan su modelo de base de datos de acuerdo con el mejor diseño seleccionado en equipo con anterioridad, con esto los estudiantes en conjunto logran seleccionar uno de sus diseños para resolver la problemática principal:

- Retoman sus análisis previos para poder trabajar en un modelado de la base de datos de forma limpia.

- Plantean ideas para incorporar nuevas entidades y relaciones que podrían servir para complementar el desarrollo del modelo que trabajan como equipo. En este paso, se sugiere que un estudiante incorpore entidades y relaciones y los otros dos las evalúen.

### Etapa 3:

En esta etapa, los estudiantes crean la base de datos utilizando sus conocimientos en base a la consulta de base de datos, elaborando lo que este modelado en base a una consulta para crear las entidades y relaciones definitivas. Además, se insertan los datos que tengan coherencia con el problema y así concluir con la consulta:

- Crean en equipo la base de datos incorporando los conocimientos de la consulta de base de datos.
- Definen los parámetros necesarios en equipo e insertan los datos correspondientes que tengan significado para la resolución del problema. Se recomienda que solo uno de ellos inserte datos y otros dos evalúen el procedimiento.

### Etapa 4:

En esta etapa, los estudiantes harán una descripción de todo lo sucedido durante el desarrollo del proyecto desde sus respectivas experiencias, generando diálogo entre ellos y solicitando que sus apuntes, notas o comentarios sean considerados, para luego hacer un informe que dé cuenta de todo el proceso para hacerlos conscientes del trabajo que se llevaron a cabo y lo puedan utilizar en otras situaciones a través de la generalización y o transferencia de la experiencia de aprendizaje. Esto hace alusión a la documentación respectiva a la posterior producción de la base de datos, con el fin de dar una retroalimentación de lo que se hizo y así explicar el proceso creativo para obtener la opción óptima para resolver el problema principal del proyecto.

También pueden responder algunas preguntas en el informe descriptivo, por ejemplo:

- ¿Cómo llegaron a resolver el problema?
- ¿Cómo fue el proceso de asignar roles en el equipo?
- ¿Qué retos afrontaron a lo largo del proyecto?
- ¿Qué aspectos del proyecto fueron significativos para ustedes?
- ¿Qué podrían mejorar de la solución obtenida para utilizarla en otros contextos?

Finalmente exponen y comparte sus soluciones con el curso o con la comunidad a través de medio digitales o de forma presencial.

## Evaluación formativa

Se recomienda que los estudiantes realicen apuntes de los análisis y bocetos de los posibles modelos que puedan ir emergiendo mientras se da el proceso de diseño. El registro de cada una de las fases del proyecto se constituye en la evidencia necesaria para realizar evaluaciones formativas por parte del profesor. Acompañado de los apuntes realizados para los desempeños específicos de cada dato o evidencia se recomienda utilizar pautas de evaluación basadas en escalas de apreciación y/o rúbricas,

que permitan a los alumnos y al profesor constatar avances, fortalezas y elementos por mejorar del proyecto.

### Evaluación sumativa

Para evaluar sumativamente se sugiere al profesor considerar tanto los resultados finales como cada una de las fases del proyecto. Para esto último, podrán utilizar las pautas aplicadas en la evaluación formativa y una pauta de evaluación de los productos finales del proyecto, como el ejemplo que se propone en Anexos.

### Difusión final

Se propone realizar un informe abarcando todo lo sucedido con el proyecto, como las evidencias del análisis, modelados y bocetos previos al resultado final del problema. Es necesario abarcar todos los detalles encontrados en la problemática principal y dar una reflexión con toda la experiencia y procesos que les permitió crear la base de datos centralizada para una empresa de distribución de alimentos, y como sus aprendizajes en cuanto al control de stock podría ser útiles en otras situaciones similares de su propia vida.

### Recursos

#### Recursos materiales

- Cuaderno o un block de notas para anotar datos o hacer bocetos de los modelos.
- Computadores para trabajar con el modelamiento y posterior consulta de datos.
- Computadores o celulares para investigar.

## Recursos

### Programas:

- Oracle Data Modeler:  
<https://www.curriculumnacional.cl/link/https://www.oracle.com/cl/database/technologies/appdev/datamodeler.html>
- Oracle SQL Developer:  
<https://www.curriculumnacional.cl/link/https://www.oracle.com/tools/downloads/sqldev-downloads.html>
- Oracle Database 18c Express Edition:  
<https://www.curriculumnacional.cl/link/https://www.oracle.com/database/technologies/xe-downloads.html>

## Anexo

### Pauta de evaluación de proyecto

Como el docente tendrá que evaluar tanto el proceso como el producto, se recomienda aplicar esta pauta teniendo a la vista la bitácora y/o apuntes del estudiante.

PAUTA PARA EVALUAR PROYECTOS				
	3	2	1	0
<b>PLANIFICACIÓN DEL PROYECTO</b>				
Descomponen el problema obteniendo la mayoría de los datos.				
Analizan el problema distinguiendo problemática principal.				
<b>DESARROLLO DEL PROYECTO</b>				
Modela base de datos que responde a la problemática.				
Prueban el funcionamiento de la consulta para la creación de tablas e inserción de datos de forma efectiva.				
Integran los datos de manera correcta, siendo coherentes para el tipo de problema.				
Cumplen con las etapas del proyecto y completan la mayoría de las tareas a tiempo.				
<b>INFORME DEL PROYECTO</b>				
Describen detalladamente el proceso de obtención del producto final.				
Explican logros y desafíos del proyecto.				
Comunican la solución al problema.				
<b>TOTAL</b>				

## Módulo Aprendizaje Basado en Resolución de Problemas

### Visión panorámica

#### Gran idea

El pensamiento algorítmico permite organizar pasos y reglas para diseñar soluciones y resolver una diversidad de problemas cuya comprensión se favorece al integrar conceptos matemáticos como variables, funciones, ecuaciones o gráficos, entre otros.

#### Objetivos de aprendizaje

**OA4.** Establecer conexiones y semejanzas entre las partes del problema para caracterizar patrones en una situación problemática dada.

**(Generalizar y Transferir)**

**OA5.** Aplicar flexiblemente las soluciones elaboradas e implementadas a otras situaciones problemáticas, integrando con honestidad diferentes ideas y puntos de vista.

**(Generalizar y Transferir)**

**OA6.** Elaborar e implementar soluciones a una situación problemática mediante la formulación de algoritmos, actuando de acuerdo con los principios de la ética. **(Pensar de forma algorítmica)**

#### Conocimientos esenciales

- Algoritmos y programación: Lenguajes, métodos y entornos de programación.
- Representación y análisis de datos.
- Impacto social de la computación: interacciones, cultura, seguridad y ética.

#### Tiempo estimado

6 semanas (12 horas)

## Propósito Módulo electivo 4

En el módulo electivo basado en el aprendizaje a través de la resolución de problemas de la asignatura Pensamiento Computacional de los Niveles 1 y 2 de Educación Media, se espera que los estudiantes logren distinguir patrones de acuerdo con un lenguaje de programación y puedan resolver diferentes situaciones problema que pueden ocurrir en la creación de programas con interfaz. Asimismo, se espera que los estudiantes comprendan que los algoritmos permiten resolver una infinidad de problemas.

Esta comprensión permitirá a los estudiantes internalizar de mejor manera las fases que puede llegar a tener un desarrollo informático y su correspondiente proceso de investigación, análisis y producción. También, los estudiantes experimentarán con diferentes algoritmos que pudieran llegar a necesitar alguna modificación para adaptarlos a lo que ellos necesiten.

Los Objetivos de Aprendizaje del módulo electivo 2 buscan desarrollar el pensar de forma algorítmica, generalizar y transferir. En particular, los estudiantes deberán aplicar sus aprendizajes en los diferentes módulos para cumplir con los desafíos propuestos, esto debido a que los retos que lleva resolver problemas y usar tecnologías son variables y responden a necesidades específicas de cada contexto.

Los Objetivos de Aprendizaje del módulo electivo 2 desarrollan las actitudes del siglo XXI del ámbito de las Maneras de pensar, Maneras de vivir el mundo y las Herramientas para trabajar promoviendo el aprovechar las herramientas y uso de la información para resolver problemas, diseñar soluciones e interactuar con sistemas en diferentes contextos de la vida diaria. Finalmente, las actividades propuestas son una oportunidad que promueve la integración de los aprendizajes desde contextos reales y cercanos para el uso de las tecnologías y la alfabetización digital.

## Problema: Una calculadora con interfaz gráfica

### Propósito

En este módulo electivo, se busca desarrollar las habilidades de pensar de forma algorítmica, generalizar y transferir, utilizando la metodología del aprendizaje basado en problemas. Para esto, se propone el contexto relacionado con la creación de una calculadora con interfaz gráfica, seleccionar los objetos que podría llegar a utilizar y la respectiva construcción en base a un lenguaje de programación determinado. Se favorecerá el desarrollo del pensamiento algorítmico descubrir su funcionamiento, modelar una interfaz coherente con el problema e identificar los objetos a utilizar.

### Preparación

Para que los estudiantes se preparen para el problema, se sugiere plantear el contexto de P.O.O. (Programación Orientada a Objetos) en el lenguaje de programación Python. Se sugiere presentar alguna definición y contexto del tema para poder guiar a los estudiantes al conocimiento previo de lo que son las interfaces gráficas en los programas:

- ¿Qué es una calculadora con interfaz gráfica?
- ¿Para qué podrían utilizarla?
- ¿Cuál es la función principal y función secundaria de una calculadora con interfaz gráfica?

En el sentido de investigar desde el problema establecido, se sugiere anotar todas las respuestas y a partir de ellas generar diferentes categorías para las características, por ejemplo, podría necesitarse algo extra en cuanto a lenguaje de programación, es algo diferente a lo ya antes establecido o la forma que podría tener. Estas categorías, en particular la forma y la percepción que se necesita algo más en concreto para crear en Python dan origen a la presentación del problema y crean formas de pensar más complejas, que luego se pueden generalizar y transferencia, especialmente en la fase de prueba de la solución.

### Presentación del problema

Para presentar el problema del módulo, se sugiere entregar el problema impreso y por medio de la escucha activa y lectura comprensiva, promover que los estudiantes registren detalles de la situación general y de su propia situación con respecto al diseño y elaboración de una calculadora con interfaz gráfica:

Para poder calcular problemas matemáticos simples, muchas compañías tienen creadas sus propias calculadoras, incluso hay aplicaciones (App) incorporadas en teléfonos móviles. Sin embargo, tengo la necesidad de diseñar y elaborar una calculadora digital con interfaz gráfica para problemas matemáticos más complejos y poner atención al proceso necesario para su programación y los requisitos para su funcionamiento ¿Qué se necesita hacer para crear una interfaz gráfica? ¿Cómo me ayudan los algoritmos para enfrentarme a la creación de un objeto? ¿Qué necesito saber para obtener un modelo visual?

### Algunas consideraciones para enfrentar el problema

- Diagnóstico de los recursos con que se cuentan para poder diseñar y programar, de manera que sea posible responder al problema.
- Apoyos en información, como información de cómo se trabaja la programación orientada a objetos y como se utilizan, de manera que se pueda trabajar objetos y no solo información secuencial.
- Preguntas previas que permiten acercarse al problema, tales como: ¿Qué se necesita para crear una interfaz en Python? ¿Cómo puedo trabajar en base a objetos? ¿Qué tipo de algoritmo me permitiría diseñar la calculadora?
- Organización para enfrentarse al problema y las posibles variaciones que se pueden hacer al problema.

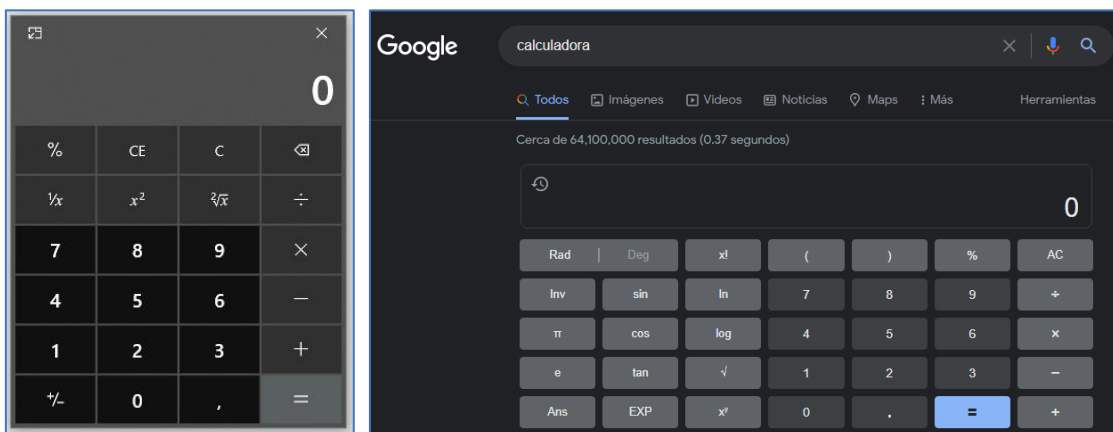
Los estudiantes asocian la situación con necesidades que pudieran darse en un contexto cotidiano o laboral, donde si bien tengo una solución computacional para una tarea, no satisface una necesidad específica. Esta etapa es importante, porque se trata esencialmente de saber cuál es la pregunta que se quiere responder y la motivación para responderla, así surgirán de manera natural los pasos hacia la solución y la necesidad del conocimiento.

### Posibles soluciones al problema

Los estudiantes experimentan con diferentes formas para trabajar en el problema, profundizan en el significado de trabajar con objetos en la programación y proponen posibles soluciones y cómo empezar a desarrollar su algoritmo para que sea satisfactorio. También, observan otras calculadoras ya creadas que se pueden ver mediante internet o en el mismo sistema operativo usado (Windows, Android, Mac, etc) para así poder identificar cómo funcionan y qué es lo que se requiere.

### Desarrollo de estrategias de acción concreta

- Pantallazo de dos calculadoras en diferentes plataformas y probar usos, secuencias, funcionamientos, tipo de datos y forma:





## Investigación

Los estudiantes indagan respecto del funcionamiento del P.O.O. (Programación Orientada a Objetos) y el cómo se pueden crear interfaces en el lenguaje de programación Python. Luego, comparten la información con sus pares en caso de ser necesario para generar mayor conocimiento común. Organizan la información en un informe descriptivo y aplican lo aprendido en una codificación de prueba para asegurar que la información adquirida servirá para resolver el problema.

La investigación que realiza el estudiante tiene el énfasis en la búsqueda de conocimientos que le ayuden a comprender la programación orientada a objetos, los algoritmos más comunes y a generalizar las relaciones que existen entre la programación, el lenguaje y la interfaz.

Las nuevas preguntas que pueden organizar esta etapa podrían ser:

- ¿Dónde sería necesario buscar datos de P.O.O e interfaces?
- ¿Para qué sirve la Programación Orientada a Objetos?
- ¿Qué herramientas servirán para lograr hacer una interfaz gráfica?
- ¿Cómo funcionan las calculadoras mediante *software*?
- ¿Cómo los aprendizajes en la asignatura ayudarán a resolver el problema?
- ¿Qué otros ejemplos puedo encontrar de programación orientada a objetos?

Estas preguntas ayudan a orientar la investigación y recopilar información necesaria para crear o modificar algoritmos posibles de usar en la programación orientada a objetos. Los apoyos de información encontrados pueden ser incorporados en el informe del proceso de obtención de la solución de manera que se genere una fuente de conocimiento individual o de equipos posibles de alimentar a través del tiempo.

## Evaluar una solución del problema

Los estudiantes evalúan la solución al problema y transfieren la misma a otras problemáticas para validarla. Se sugiere utilizar la información obtenida en el proceso de investigación para analizar la solución y, a su vez, considerar si es necesario obtener más información para luego dar paso a un proceso de comparación (semejanzas y diferencias) y lo factible que puede llegar a ser usar el modelo o una modificación de este para resolver el problema.

Se sugiere presentar el desarrollo de la situación, por ejemplo: qué método se utiliza para poder obtener una interfaz gráfica y su funcionamiento. El método para utilizar se puede obtener de la página web <https://www.curriculumnacional.cl/link/https://docs.python.org/es/3/library/tk.html>, Aquí se ven algunos ejemplos de las funcionalidades que puede llegar a tener el incorporar **un método** en Python.

Métodos	Descripción
Grid	El método grid nos permite posicionar los widgets en una celda en específico. Indicamos la celda usando el índice de fila y columna correspondiente; el ancho y la altura de cada celda son configurables. Además, un widget puede ocupar varias celdas si lo deseamos. Usando grid podemos crear fácilmente interfaces gráficas de usuario tipo formulario.
Label	El método label son textos estáticos, de ahí que se llame label o etiqueta de texto.
Entry	Una caja de texto que permite al usuario ingresar cualquier texto de una línea.
Text	Una caja de texto que permite al usuario ingresar cualquier texto en varias líneas.

Además de describir las funcionalidades que pueden llegar a ejercer los métodos antes mencionados, es requerido investigar **los modificadores u opciones** que pueden llegar a tener.

Opciones	Descripción
bg	El color de fondo predeterminado del widget de texto.
bd	El ancho del borde alrededor del widget de texto. El valor predeterminado es 2 píxeles.
font	La fuente predeterminada para el texto insertado en el widget.

Se sugiere presentar la definición de las opciones en un cuadro y describir los métodos en orden, para así inscribir cuáles opciones sirven en cada método. Además, se pedirá una investigación completa de las funcionalidades P.O.O. y para qué sirven en el uso de la programación, por lo tanto, es necesario dar ejemplos y descripciones de lo que se necesita para trabajar con dicha información.

Para finalizar, se requiere de un análisis de lo propuesto en la investigación, esto quiere decir, que se requiere una conclusión y especificación de lo que se utilizará para desarrollar la solución al problema: calculadora con interfaz gráfica, con el fin de representar un pensamiento algorítmico previo a la producción del programa.

### Elaborar y ajustar la investigación

Los estudiantes describen diferentes funcionalidades, métodos, opciones y widgets capaces de proporcionar lo necesario para crear una calculadora en el lenguaje de programación de Python. Para esto, se elabora una investigación y descripción de los diferentes métodos y opciones que pueden incorporarse y adaptarse al problema principal:

- La consideración de diferentes fuentes para poder crear interfaces gráficas debería quedar expresada según la compatibilidad y facilidad de uso que se podría llegar a tener.

- Las definiciones obtenidas se evalúan bajo la pregunta: ¿Sirve esta información para cumplir con la solución?, ¿sirve esta información y sus definiciones para lograr lo requerido?
- La evaluación de la investigación debe describir una serie de funcionalidades que se puedan utilizar en el lenguaje de programación correspondiente y así tener una selección de lo que se podría usar.

## Comunicar

Los resultados obtenidos son comunicados y comparados entre ellos, por ejemplo: los diferentes métodos que se pueden utilizar para diseñar interfaces gráficas en Python, los modificadores u opciones que estos pueden tener, el cómo se podría diseñar una calculadora con interfaz gráfica, entre otros. También se puede reflexionar sobre las estrategias utilizadas en el proceso de investigación, sobre los algoritmos que se podrían utilizar en el proceso creativo, la diferencia de los datos obtenidos, o la posibilidad de transferir los conocimientos obtenidos a otras situaciones.

Algunas de las preguntas que pueden guiar la conversación son:

- ¿Qué tiene en común los métodos y funciones?
- ¿Qué pregunta nueva o diferente responde el modelo creado respecto de los investigados?
- ¿Qué métodos son utilizadas en mi modelo en comparación con otros?

Se sugiere comunicar la información y las sugerencias utilizando alguna de estas posibilidades:

- Presentación
- Informe
- Video
- Simulación

## Orientaciones al docente

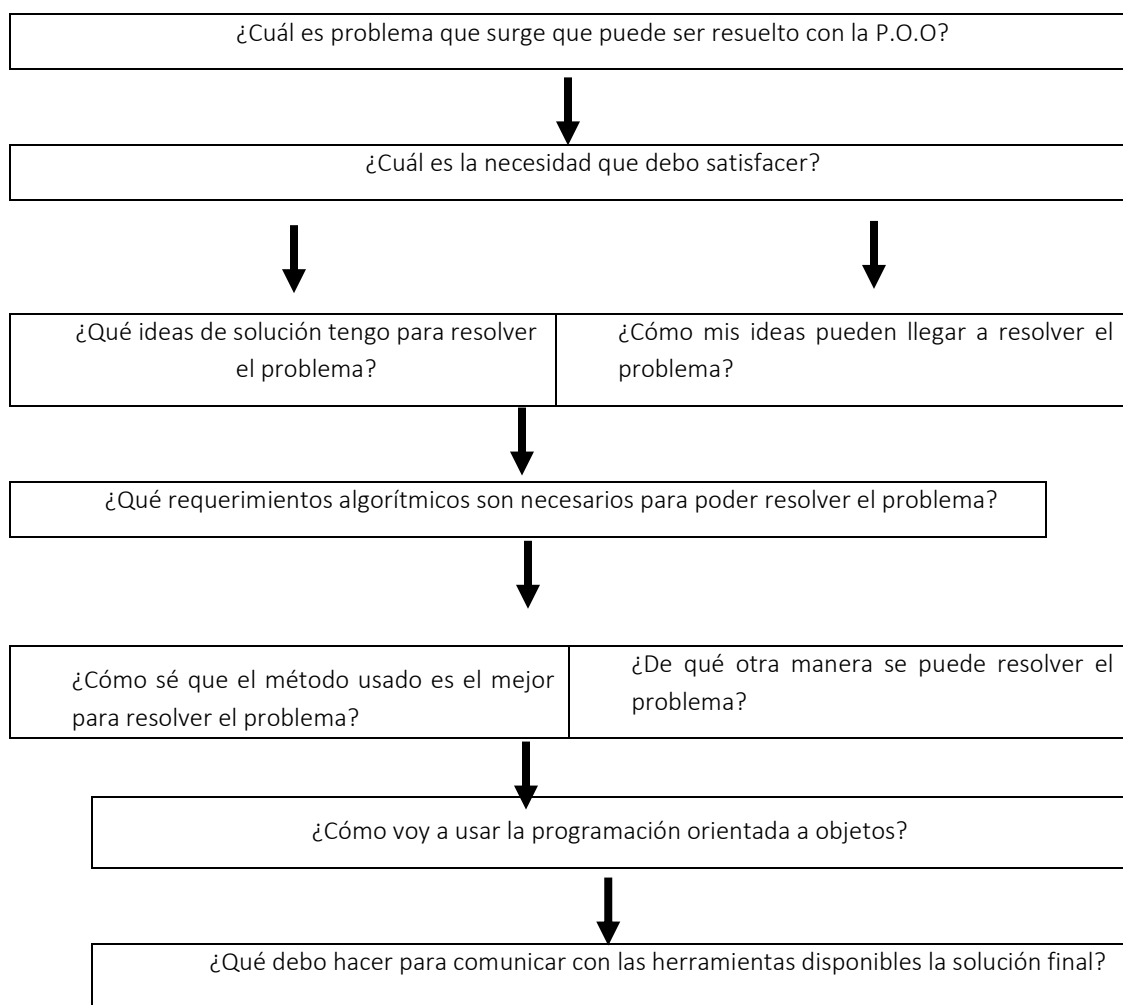
**Para unificar conceptos disciplinares:** se sugiere precisar las opciones que pueden variar requiriendo de cada información adquirida por el estudiante. Cabe resaltar que el método de “Tkinter” no es el único método conocido para crear interfaces gráficas dentro de Python, también existe “PySimpleGUI”. Además, se puede usar un sistema de capas junto a HTML para crear una interfaz web junto a “Django”. El límite puede variar dependiendo de lo que se busque y lo que se quiera hacer dentro del lenguaje, pero es seguro que el método más adecuado para esta experiencia en particular es “Tkinter” por su fácil acceso y funcionalidades.

El uso que se les puede dar a cada una de ellas dependerá de la investigación que se haga al respecto y con ello la comprensión que se pueda dar a la información en general y particular. Es esencial que el docente tenga su propia investigación para orientar a los estudiantes.

**Para focalizar el desarrollo de habilidades:** aunque la actividad desarrolla varias habilidades del lenguaje de programación y el cómo se pueden incorporar métodos externos que fueron diseñados para este sistema, se sugiere focalizar el desarrollo en solo una de ellas. Para esto se puede tener presente la siguiente estrategia interrogativa de manera de enfocar el desarrollo de la habilidad pensar de forma algorítmica.

- ¿Cuál es problema que surge que puede ser resuelto con la P.O.O?
- ¿Cuál es la necesidad que debo satisfacer?
- ¿Qué ideas de solución tengo para resolver el problema?
- ¿Cómo mis ideas pueden llegar a resolver el problema?
- ¿Qué requerimientos algorítmicos son necesarios para poder resolver el problema?
- ¿Cómo sé que el método usado es el mejor para resolver el problema?
- ¿De qué otra manera se puede resolver el problema?
- ¿Cómo voy a usar la programación orientada a objetos?
- ¿Qué debo hacer para comunicar con las herramientas disponibles la solución final?

Además, poner a disposición de los estudiantes un organizador gráfico que puede ser utilizado como bitácora para el proceso de solución al problema:



## ANEXO 1

### BIBLIOGRAFÍA

Araya, R., Isoda, M., González, O., e Inprasitha, M. (2019). A Framework for Computational Thinking for the Transition to a Super Smart Society. APEC InMside Report.

Artecona, F., Bonetti, E., Darino, C., Mello, F., Rosá, M., y Scópise, M. (2017). Pensamiento Computacional. Un Aporte para la Educación de Hoy. Uruguay: Fundación Telefónica. Recuperado de <https://www.curriculumnacional.cl/link/https://www.fundaciontelefonica.cl/arte-y-cultura-digital/publicaciones-listado/pagina-item-publicaciones/itempubli/618/>

Basogain, X., Olabe, J., & Del Rosario, R. (2016). PC-01: Introduction to computational thinking: Educational technology in primary and secondary education - IEEE Conference Publication. Recuperado de <https://www.curriculumnacional.cl/link/https://ieeexplore.ieee.org/document/7751816/citations?tabFilter=papers>

CAS (2015). Pensamiento Computacional. Guía para profesores. Computing At School. Recuperado de <https://www.curriculumnacional.cl/link/http://www.codemas.org/wp-content/uploads/2016/04/Pensamiento-computacional-Guía-para-profesores.pdf>

Fábrega, R., Jara, M., Carreño, E., & Osorio, M. (2016). Enseñanza de Lenguajes de Programación en la Escuela ¿Qué están haciendo en otros países? Santiago: Fundación Telefónica. Recuperado de [https://www.curriculumnacional.cl/link/http://www.fundaciontelefonica.cl/wp-content/uploads/descargas/1504030044-Ense%C3%B1anza%20de%20Lenguajes%20de%20Programaci%C3%B3n%20en%20otros%20pa%C3%A](https://www.curriculumnacional.cl/link/http://www.fundaciontelefonica.cl/wp-content/uploads/descargas/1504030044-Ense%C3%B1anza%20de%20Lenguajes%20de%20Programaci%C3%B3n%20en%20otros%20pa%C3%A1ses.pdf)  
Dses.pdf

Grover, S. & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. Educational Researcher. Vol. 42, Nº 1, pp. 38-43.

ISTE (2011). Operational definition of computational thinking for K-12 education. NSF, CSTA, ISTE. Recuperado de [https://www.curriculumnacional.cl/link/https://cdn.iste.org/www-root/Computational\\_Thinking\\_Operational\\_Definition\\_ISTE.pdf?\\_ga=2.79189782.305816076.1642012397-617136009.1642012397](https://www.curriculumnacional.cl/link/https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf?_ga=2.79189782.305816076.1642012397-617136009.1642012397)

Jara, M., & Fábrega, R. (2017). Bases para la implementación de la Enseñanza de la Programación en el Sistema Escolar: Exploración de Estrategias Públicas y Privadas. Santiago: Fundación Telefónica. Recuperado de <https://www.curriculumnacional.cl/link/https://www.fundaciontelefonica.cl/arte-y-cultura-digital/publicaciones-listado/pagina-item-publicaciones/itempubli/641/>

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? Computers in Human Behavior, 41, 51–61. Recuperado de <https://www.curriculumnacional.cl/link/https://doi.org/10.1016/j.chb.2014.09.012>

Ministerio de Educación de Chile (2019). Bases curriculares 3° y 4° medio: Plan de formación general, plan de formación diferenciada humanístico-científico. Santiago. Recuperado de [https://www.curriculumnacional.cl/614/articles-91414\\_bases.pdf](https://www.curriculumnacional.cl/614/articles-91414_bases.pdf)

Ministerio de Educación, UCE. (2019) (2019) ¿Qué es ABP? en: <https://www.curriculumnacional.cl/portal/Secciones/ABP-Chile-aprende-por-proyectos/134607:Que-es-ABP>

Ministerio de Educación, UCE. (2019) Metodología de aprendizaje basado en proyectos en: <https://www.curriculumnacional.cl/portal/Curso/Tecnico-Profesional/3-Medio-TP/140166:Metodologia-de-aprendizaje-basado-en-proyectos>

Ministry of Education. (2017). The New Zealand curriculum. Technology Learning Area. Recuperado de <https://www.curriculumnacional.cl/link/http://nzcurriculum.tki.org.nz/The-New-Zealand-Curriculum>

Resnick, M. (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. C&C '07 Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition. Washington D. C.

Rico Lugo, M. J. y Bosagain Olabe X. (2018). Pensamiento computacional: rompiendo brechas digitales y educativas. EDMETIC, Revista de Educación Mediática y TIC, 7(1), 26-42, doi: <https://www.curriculumnacional.cl/link/https://doi.org/10.21071/edmetic.v7i1.10039>

Tedre, Matti; Denning, Peter J. (2016). The Long Quest for Computational Thinking. Proceedings of the 16th Koli Calling Conference on Computing Education Research, November 24-27, 2016, Koli, Finland: pp. 120-129.

Vivian, R., Falkner, K., & Falkner, N. (2014). Addressing the challenges of a new digital technologies curriculum: MOOCs as a scalable solution for teacher professional development. Research in Learning Technology, 22 (Special Issue: Riding Giants: How to innovate and educate ahead of the wave), 1–19.

Wing, J. (2006). Computational Thinking. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. Communications of the ACM, 49(3), 33-35. Recuperado de <https://www.curriculumnacional.cl/link/https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>

Wing, J. (primavera, 2011). Computational thinking - What and why? The Link, (6), pp. 20-23. Recuperado de [https://www.curriculumnacional.cl/link/https://www.cs.cmu.edu/sites/default/files/11-399\\_The\\_Link\\_Newsletter-3.pdf](https://www.curriculumnacional.cl/link/https://www.cs.cmu.edu/sites/default/files/11-399_The_Link_Newsletter-3.pdf)

Zapata, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. RED-Revista de Educación a Distancia. Vol. 46, Nº 4. Recuperado de <https://www.curriculumnacional.cl/link/https://revistas.um.es/red/article/view/240321/183001>

## ANEXO 2

### SITIOS WEB

#### Ejercicio IMC:

[https://www.curriculumnacional.cl/link/https://www.cdc.gov/healthyweight/spanish/assessing/bmi/adult\\_bmi/index.html](https://www.curriculumnacional.cl/link/https://www.cdc.gov/healthyweight/spanish/assessing/bmi/adult_bmi/index.html)

#### Funciones de SQL Developer Oracle:

<https://www.curriculumnacional.cl/link/https://www.techonthenet.com/oracle/index.php>

#### Funciones de SQL Data Modeler Oracle:

[https://www.curriculumnacional.cl/link/https://docs.oracle.com/cd/E15276\\_01/doc.20/e13677.pdf](https://www.curriculumnacional.cl/link/https://docs.oracle.com/cd/E15276_01/doc.20/e13677.pdf)

#### Funciones de Python:

<https://www.curriculumnacional.cl/link/https://www.w3schools.com/python/default.asp>

#### Funciones de PSeint:

[https://www.curriculumnacional.cl/link/https://www.itson.mx/oferta/isw/Documents/guia\\_pseint\\_2016.pdf](https://www.curriculumnacional.cl/link/https://www.itson.mx/oferta/isw/Documents/guia_pseint_2016.pdf)

<https://www.curriculumnacional.cl/link/http://pseint.sourceforge.net/index.php?page=ejemplos.php>

#### Programación por bloques:

<https://www.curriculumnacional.cl/link/https://studio.code.org>

<https://www.curriculumnacional.cl/link/https://scratch.mit.edu/>

#### Programación y diseño:

<https://www.curriculumnacional.cl/link/https://www.tinkercad.com/>

<https://www.curriculumnacional.cl/link/https://www.rstudio.com/>

#### Fundaciones para el desarrollo tecnológico

<https://www.curriculumnacional.cl/link/https://cuantrix.mx/>

<https://www.curriculumnacional.cl/link/https://www.kodea.org/>

<https://www.curriculumnacional.cl/link/https://www.iste.org/>

<https://www.curriculumnacional.cl/link/https://www.csteachers.org>