



ESPECIALIDAD PROGRAMACIÓN

SECTOR TECNOLOGÍA Y COMUNICACIONES

4° AÑO EDUCACIÓN MEDIA

MALETÍN DIDÁCTICO

PROGRAMACIÓN ORIENTADA A OBJETOS



EDITORIAL

El proyecto fue desarrollado por un equipo profesional interdisciplinario de la Universidad de La Frontera (UFRO), compuesto por especialistas, docentes TP, académicos del área de programación, pedagogos especialistas en currículum, evaluación y educación técnico profesional.

Coordinador de Proyecto

Pablo Fuentes Iturra.

Equipo Pedagógico y Curricular

Pablo Álvarez Gómez, Fresia Contreras Armijo, Karina Uribe Mansilla y Juan Vergara Palma.

Equipo Disciplinar

Matías Yañez Pohl.

Revisión General

Loreto Cárdenas Baeza.

Diseño Gráfico

Daniela Silva Hidd.

CONTENIDO

1.	CONTEXTUALIZACIÓN	5
	1.1. Estructura de Maletín Didáctico del Módulo	5
2.	RUTA DE APRENDIZAJE	6
3.	PROPUESTAS DE ACTIVIDADES DE APRENDIZAJE CONTEXTO PRESENCIAL Y REMOTO	8
	3.1. Sugerencias Generales	9
4.	ANEXO 1.OTRAS ESTRATEGIAS DE EVALUACIÓN	23

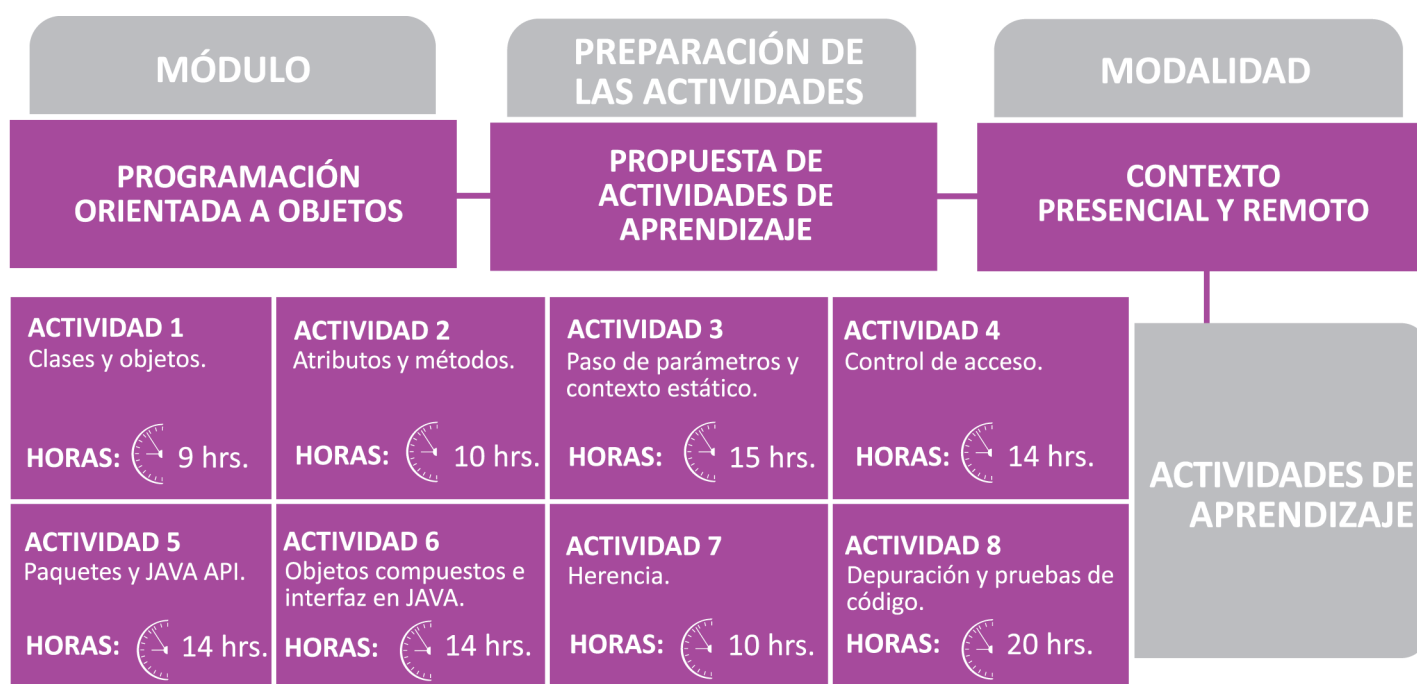
1. CONTEXTUALIZACIÓN

Este módulo de **228 horas**, da inicio al aprendizaje práctico de los diferentes aspectos de programación orientada a objetos, simulando un ambiente de trabajo propicio para pruebas y desarrollo de aplicaciones, de acuerdo a las necesidades de la industria. En este contexto, para fortalecer la propuesta curricular y pedagógica del módulo se integra un nuevo aprendizaje esperado vinculado al desarrollo de programas de mediana complejidad, y que será abordado desde la propuesta de actividades de aprendizajes.

El objetivo principal en esta etapa es que los y las estudiantes puedan resolver situaciones propias de un desarrollo bajo estos conceptos. Para ello, tendrán que concretar soluciones informáticas, ampliando la lógica resolutoria de casos de negocios empresariales en diversos contextos de programación.

1.1. ESTRUCTURA DE MALETÍN DIDÁCTICO DEL MÓDULO

Este maletín didáctico está compuesto por:



2. RUTA DE APRENDIZAJE

El propósito de esta ruta de aprendizaje es visibilizar los Objetivos de Aprendizaje **(OA)**, Aprendizajes Esperados **(AE)** y Criterios de evaluación **(CE)** involucrados en la propuesta formativa del módulo, destacando aquellos que son abordados en esta modernización y que en su conjunto contribuyen al desarrollo de las competencias esperadas para los y las estudiantes. En este sentido, se efectúa una abreviación de cada uno de los **AE** y **CE**, y se integran dos íconos para distinguir la selección de estos en la modernización y la articulación con el módulo de Emprendimiento y empleabilidad.

Esta propuesta integra un Aprendizaje Esperado nuevo y tiene el propósito de introducir a conceptos básicos de la programación orientada a objetos **(AE5)**, abordado principalmente a través de siete actividades usando metodología basada en retos, para luego incursionar en técnicas de depuración y pruebas de código **(AE1 y AE2 respectivamente)** que son tratados de manera conjunta en una actividad que invita a los estudiantes a programar, probar y depurar sus líneas de código en una competencia grupal usando herramientas basadas en la nube. Finalmente, se introduce una capa de calidad **(AE1 y AE2 respectivamente)** en las primeras siete actividades para profundizar la construcción de los aprendizajes y dar cuenta de algunas etapas del proceso de desarrollo de software como lo son la construcción y validación de un producto tecnológico.

Una vez completadas estas ocho actividades se invita a los estudiantes a realizar un proyecto de programación que permita aplicar los conceptos aprendidos anteriormente, creando una solución tecnológica con miras a los objetivos de desarrollo sostenible de las Naciones Unidas. Para esto se articula formativamente con el módulo de Emprendimiento y empleabilidad.

Si bien los últimos dos aprendizajes esperados no están considerados en el maletín didáctico, se entregan orientaciones en la propuesta de actividades para la integración al proyecto de programación descrito anteriormente.



PROGRAMACIÓN ORIENTADA a OBJETOS

Construir aplicaciones computacionales basadas en programación orientada a objetos, de manera de cumplir con las exigencias técnicas y de los usuarios.



APRENDIZAJES ESPERADOS

AE

AE5

AE1

AE2

AE3

AE4

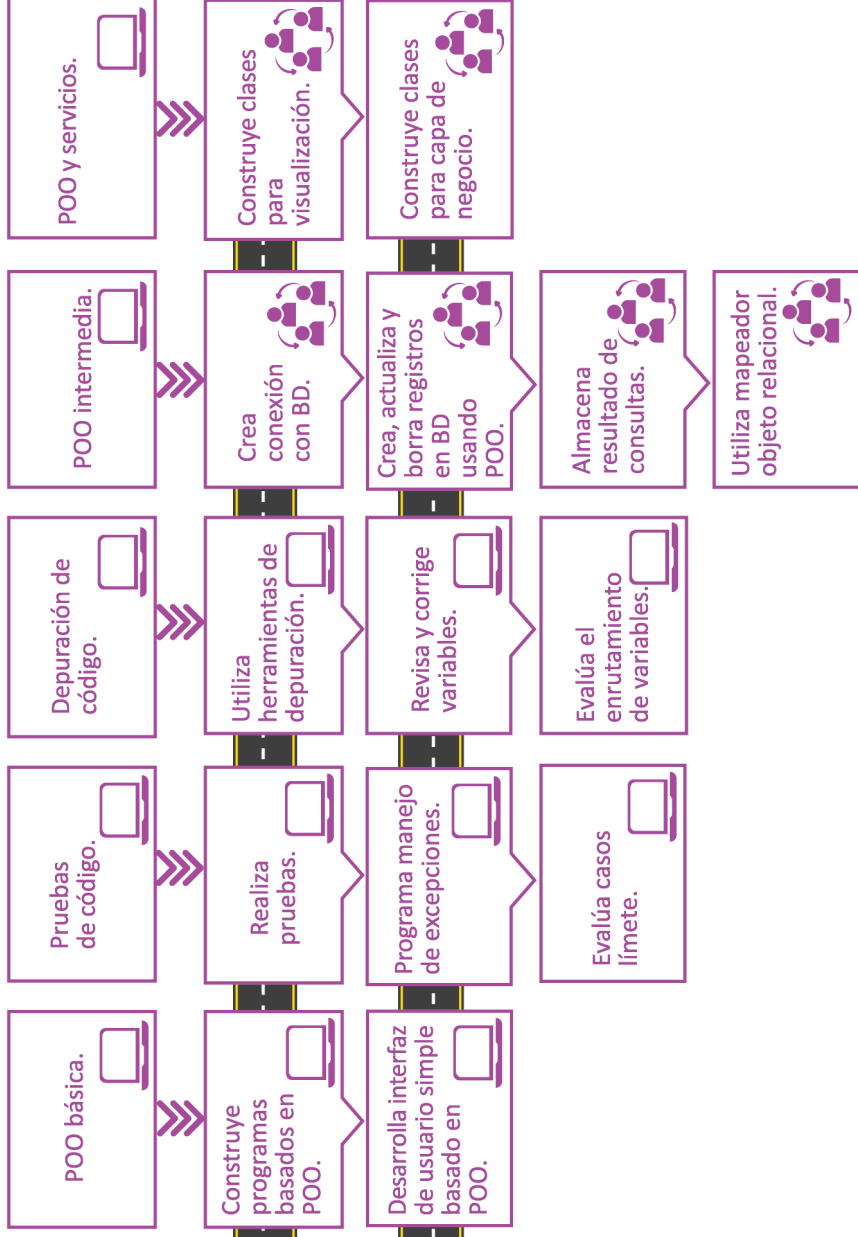
CRITERIOS de EVALUACIÓN

CE



Modernizados

Emprendimiento y Empleabilidad



3. PROPUESTAS DE ACTIVIDADES DE APRENDIZAJE CONTEXTO PRESENCIAL Y REMOTO

La siguiente propuesta de actividades tiene como propósito evidenciar los Objetivos de aprendizaje técnicos y genéricos, Aprendizajes Esperados (AE), Criterios de Evaluación (CE) y otros elementos esenciales que se considerarán en el abordaje de este módulo modernizado de Programación Orientada a Objetos (POO). Posteriormente se presentan ocho actividades de aprendizaje, tanto para el contexto presencial como remoto.

De las 228 horas que dispone el módulo, se propone una innovación formativa de 106 horas, que permitirá fortalecer el logro de los Objetivos de Aprendizaje. Para la modernización de este módulo, se considera el desarrollo de aplicaciones, videojuegos y experiencias inmersivas como realidad virtual/aumentada.

En una primera instancia, se abordan los conceptos fundamentales de POO y luego, los y las estudiantes podrán progresar a evaluar códigos de programación y corregirlos. Lo anterior, en base a la metodología de Aprendizaje Basado en Retos (ABR). La actividad final, que está articulada con el módulo de Emprendimiento y empleabilidad, permite el desarrollo de un proyecto integrador abordando los contenidos modernizados mencionados anteriormente.

Las actividades propuestas sugieren en forma general la articulación también con otros módulos de la especialidad como, Desarrollo de aplicaciones web y asignaturas de formación general como inglés, con la finalidad de interiorizarse en los procesos de investigación y lectura de documentación técnica de programación que se encuentra en internet. En particular, se dará énfasis en las Soluciones tecnológicas, basadas en el paradigma de la programación orientada a objetos (desarrollo de videojuegos, AR/VR y web).

Como parte del fortalecimiento del módulo, el AE5, será abordado en las actividades 1, 2, 3, 4, 5, 6 y 7 junto a sus criterios de evaluación (CE 5.1 para las actividades 1, 2 y 4; y CE 5.1 y CE 5.2 para las actividades 3, 5, 6 y 7). Por su parte, los AE1 y AE2 en conjunto con sus criterios de evaluación CE 1.1, 1.2, 1.3 y CE 2.1, 2.2, 2.3, serán abordados en una primera etapa mediante la actividad 8, para luego reforzar los aprendizajes en las actividades 2, 3, 4, 5, 6 y 7.

En este sentido, el proceso de aprendizaje y evaluación contempla distintas estrategias que deben ser contextualizadas a cada aula y que consideran distintos instrumentos de evaluación para adaptar según corresponda. Es fundamental concebir que estos instrumentos deben ser compartidos, consensuados y comprendidos con las y los estudiantes antes de iniciar cada actividad, ya que cumplen el propósito de orientar o guiar los aprendizajes, autoevaluar y/o coevaluar los desempeños, lo que permite movilizarse hacia una autorregulación de los mismos.

Es importante destacar que la retroalimentación debe efectuarse constantemente para lograr un mayor impacto en la ejecución de las actividades y los errores deben ser utilizados como fuente aprendizaje. Adicionalmente, es necesario que la autoevaluación y la coevaluación sean entendidas como espacios de reflexión crítica respecto a lo construido a lo largo de las actividades, visibilizando las oportunidades de mejora en este proceso formativo. La calificación dependerá de cada docente y contexto de aula.

3.1. SUGERENCIAS GENERALES

A continuación se detallan algunas sugerencias para considerar en las distintas actividades tanto en un contexto presencial como remoto:

- Para mayor profundidad en las actividades se recomienda abordarlas por partes y distribuir la cantidad de horas por cada una, según su contexto. Acordar fechas previas y tiempos de entrega de productos para resolver dudas y retroalimentar el proceso.
- Complementar la clase conversando con los y las estudiantes sobre las últimas noticias relacionadas con tecnología e industria 4.0. (Recursos: Link 1, Link 2, Link 3, Link 4).
- Conversar sobre experiencias de los temas tratados, acercando el conocimiento a situaciones aplicables en el campo laboral.
- Para la organización de grupos de trabajo de los y las estudiantes (actividades 1, 2, 3, 4, 5 y 6), se invita a usar Discord para comunicación en tiempo real.
- Utilizar distintos instrumentos de evaluación para el aprendizaje, como por ejemplo los sugeridos en el Anexo 1 como parte de las estrategias de evaluación para favorecer aprendizajes profundos en los y las estudiantes. Estos, fueron adaptados de las orientaciones y estrategias evaluativas de la Unidad de Currículum y Evaluación del Ministerio de Educación. Revisar más recursos en el siguiente link:
https://www.curriculumnacional.cl/portal/Documentos-Curriculares/Evaluacion/#recuadros_articulo_7330_5
- Verificar si los y las estudiantes cuentan con los medios para realizar las actividades solicitadas, en particular acceso a internet, para determinar plataformas accesibles con las cuales trabajar y que permitan conocer la retroalimentación simultánea, como es el caso de herramientas GSuite, así como también, que permitan alojar archivos en una carpeta compartida para que todos tengan acceso.
- Alojar los recursos de cada actividad en una carpeta compartida o plataforma acordada con los y las estudiantes. Según corresponda, indicar que las guías pueden ser trabajadas como documentos compartidos en la nube, dando la posibilidad a los y las estudiantes que puedan ir dejando sus comentarios o dudas y en forma simultánea ir revisándolos y retroalimentando.
- Para la actividad 8, presentar un ejemplo que explique la secuencia de pasos postulada en la **“Presentación de depuración y pruebas de código”**. Explicar las herramientas de depuración con un código que presente fallas para ver las potencialidades de la depuración.
- Integrar herramientas digitales que promuevan la participación de los y las estudiantes como:
 - a. Mentimeter (<https://www.mentimeter.com/>). Se pueden generar preguntas y que al ser respondidas según el elemento seleccionado sean visualizadas en forma gráfica y con datos cuantitativos. Para tener acceso gratuito, registrarse.
 - b. Jamboard (pizarra digital de GSuite). Se pueden registrar por ejemplo las ideas principales o conclusiones, insertar imágenes, entre otros. Para su descarga hacer clic en <https://play.google.com/store/apps/details?id=com.google.android.apps.jam&hl=es&gl=US> o bien utilizarla desde las opciones de aplicaciones del correo electrónico asociado a una cuenta Google.
 - c. Genially (<https://www.genial.ly/es>) o Canva (<https://www.canva.com/>). Puede ser una opción por ejemplo para dinamizar la muestra de fotografías a través de alguna plantilla de la plataforma o generar contenidos interactivos. Para tener acceso gratuito, registrarse.
 - d. Padlet (<https://es.padlet.com/>). Puede ser una opción para presentar información, sintetizar, hacer tableros, documentos interactivos, entre otros. Para tener acceso gratuito, registrarse.

3.1. SUGERENCIAS GENERALES

- Hacer hincapié en la importancia de comprender el idioma inglés en la programación, por ejemplo, al entender los mensajes de error que lanza el IDE de desarrollo. Se sugiere, en la clase de inglés, la traducción de **Link 5**, abordando **IN4M AC OF 01** como Objetivo de Aprendizaje.
- Se sugiere efectuar espacios de reflexión sobre las prácticas pedagógicas en conjunto con los pares y jefe de UTP o equipo directivo, enfocando estas instancias con un propósito formativo. Utilice el siguiente recurso **Pauta de Reflexión_Docente.docx**.

A continuación se detallan las distintas propuestas de actividades para el fortalecimiento del módulo. Cabe señalar, que las dimensiones de las competencias a desarrollar son los conocimientos (**saber**), habilidades (**saber hacer**) y actitudes (**saber ser**), y que a partir de su análisis didáctico se dan a conocer los elementos de competencia a considerar.

ESPECIALIDAD	PROGRAMACIÓN					NIVEL	4° MEDIO	
NOMBRE DEL MÓDULO	PROGRAMACIÓN ORIENTADA A OBJETOS					TOTAL DE HORAS	106	
ELEMENTO NUEVO	OA	No aplica	AE	AE5	CE	5.1 - 5.2	RECURSO	Aplicaciones, Videojuegos, AR/VR

OBJETIVO DE APRENDIZAJE TÉCNICO	OA4 Construir aplicaciones computacionales basadas en programación orientada a objetos, de manera de cumplir con las exigencias técnicas y de los usuarios.						
--	---	--	--	--	--	--	--

APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	OBJETIVOS DE APRENDIZAJE GENÉRICOS
AE5 Construye un programa de mediana complejidad sobre la base de los principios de la programación orientada a objetos, articulando los aprendizajes con el Módulo de Programación y Base de Datos.	5.1 Desarrolla programa individual o de forma colaborativa, basado en programación orientada a objetos, considerando diversos requerimientos funcionales para la solución de un problema dado, aplicando buenas prácticas de la industria.	A - B - C - G - H
	5.2 Desarrolla interfaz de usuario simple, basado en programación orientada a objetos, para navegar a través del programa, considerando requerimientos circunstanciales de un determinado problema.	A - B - C - G - H
AE1 Construye unidades de prueba para verificar el correcto funcionamiento de la codificación realizada, de acuerdo a exigencias técnicas de confiabilidad.	1.1 Realiza pruebas para detectar problemas previos a la codificación de una unidad de software con una herramienta de software disponible en el mercado.	A - C - H
	1.2 Construye una unidad de prueba para detectar excepciones en un entorno de lenguaje con una herramienta compatible que permita desarrollar, configurar e Implementar aplicaciones en función del lenguaje utilizado.	A - C - H
	1.3 Construye una unidad de prueba para detectar casos límites, en un entorno de lenguaje, con una herramienta compatible, propia del software utilizado como marco de trabajo.	A - C - H
AE2 Detecta y corrige errores de codificación, analizando el comportamiento del código de programación, de acuerdo a especificaciones y manuales de referencia.	2.1 Utiliza la herramienta de depuración de un entorno de desarrollo, para revisar la codificación de clases programadas.	A - C - H
	2.2 Revisa y corrige las variables, del programa desarrollado para la solución de un caso, indagando en las funciones internas de acuerdo a protocolos de revisión.	A - C - H
	2.3 Revisa las variables de una función de clases programadas, utilizando opciones del lenguaje empleado, para asegurar su correcto funcionamiento.	A - C - H

ESPACIOS DE APRENDIZAJES

ALTERNANCIA	Empresa	No Aplica	ARTICULACIÓN	Módulo	M9: Emprendimiento y empleabilidad
	IES	No Aplica		Especialidades	No Aplica
				Formación General	Sugerencia general

NOMBRE DE ACTIVIDAD	PROGRAMACIÓN ORIENTADA A OBJETOS		
COMPETENCIAS Análisis didáctico	CONOCIMIENTOS	HABILIDADES	ACTITUDES
	<ul style="list-style-type: none"> • Clases y objetos en POO. • Atributos y métodos en POO. • Paso de parámetros y contexto estático. • Control de acceso en POO. Paquetes en Java. • Objetos compuestos e interfaces en Java. • Herencia en Java. • Técnicas de depuración de código Y Pruebas unitarias de código. 	<ul style="list-style-type: none"> • Apropiación de lenguaje técnico. • Identificación de clases, objetos, atributos y métodos en programación orientada a objetos (POO). • Creación de clases definiendo sus atributos y métodos. • Identificación y corrección de errores de código. • Creación de pruebas unitarias en Java. • Identificación de paso por valor y por referencia. • Identificación y construcción de elementos estáticos en POO. • Identificación y construcción de elementos para controlar el acceso a elementos en POO. • Identificación, construcción y utilización de paquetes en Java. • Creación e importación de paquetes en Java. • Identificación y creación de objetos compuestos e interfaces en Java. • Identificación y creación de clases heredadas en Java. 	<ul style="list-style-type: none"> • Utiliza material relacionado con el trabajo. • Trabajo prolijo cumpliendo plazos establecidos. • Participa en diversas situaciones del aprendizaje. • Trabajo en equipo.
METODOLOGÍA SELECCIONADA	APRENDIZAJE BASADO EN RETOS¹		
AMBIENTES DE APRENDIZAJE	CONTEXTO PRESENCIAL: Desarrollar actividad en sala o laboratorio de computación. CONTEXTO REMOTO: Plataforma virtual o software de video conferencia.		
ETAPAS	DESCRIPCIÓN DE LAS ACTIVIDADES DE APRENDIZAJE		
1 Preparación de la Actividad	<ul style="list-style-type: none"> • Identifica el contexto de sus estudiantes y ritmos de aprendizajes, junto con adaptar los recursos pedagógicos disponibles para preparar las clases (según corresponda, acuerda algún medio o plataforma virtual para el desarrollo de las clases). • Prepara de IDE de desarrollo (IntelliJ Idea, NetBeans, Visual Studio Code, Eclipse, entre otros). • Cuenta con el set de materiales que se proponen en las páginas 21 y 22. 		

¹Basado en el Marco metodológico del Aprendizaje Basado en Retos de Apple (2011), recuperado en Observatorio de Innovación Educativa del Tecnológico de Monterrey <https://observatorio.tec.mx/edutrendsabr>

ETAPAS

DESCRIPCIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

2

Ejecución

Docente

- Según corresponda, inicia clases vía software de videoconferencia y recomienda mantener micrófonos en silencio y activarlos cuando algún estudiante desee hablar.
- Crea un ambiente propicio para el aprendizaje, generando un clima de respeto y participación. En conjunto acuerdan normas de convivencia.
- Da a conocer los objetivos o propósitos de cada clase y/o actividad y efectúa preguntas respecto a su comprensión.
- Proyecta las presentaciones en sala de clases o comparte pantalla en el software de videoconferencia, para el inicio y desarrollo de cada actividad.
- Da a conocer que para la ejecución de las actividades se plantean **al menos 8**, vinculadas a adquirir conocimientos, habilidades y actitudes sobre programación orientada a objetos. En este sentido las actividades tanto teóricas como prácticas son:

Actividad 1. Clases y Objetos

- Da inicio al módulo de programación orientada a objetos, dando a conocer la importancia que tiene en el desarrollo de software este paradigma de computación. Deja abierta la pregunta sobre qué es un paradigma para abordarla más adelante.
- Mediante el recurso **1.1_Presentación clases y objetos_Docente.pptx** realiza lo siguiente:
- Activa el aprendizaje reforzando los conocimientos, habilidades y actitudes alcanzadas en programación estructurada por medio del **video 1**, considerado en la **diapositiva 3** de la presentación <https://www.youtube.com/watch?v=STVXkRO4LZY>.
- Presenta el ejercicio **“Desafío 0: Divisas”** como evaluación diagnóstica, guiando los aprendizajes según el recurso **1.3_Instrumento de evaluación de clases y objetos desafío 0_Estudiante.docx**. Luego invita a una conversación abierta sobre eventuales soluciones al problema planteado. Entrega retroalimentación sobre buenas prácticas en programación.
- Asegura la comprensión de **“Introducción a la POO”** e invita a participar activamente en conversación en torno a la temática.
- Analiza en conjunto el **video 2** relacionado con los conceptos que se abordarán en la clase, considerado en la **diapositiva 10** de la presentación <http://www.youtube.com/watch?v=Nka4JSBgf7I>
- Introduce los conceptos de paradigma y formula preguntas como: ¿Conocen el concepto de paradigma? ¿Me podrían dar algún ejemplo? ¿Alguien ha escuchado qué es la POO? Entabla una conversación sobre las respuestas de a estas preguntas e introduce el concepto de objeto.
- Muestra ejemplos donde la programación estructurada presenta dificultades para construir soluciones. Se formula(n) la(s) pregunta(s): ¿Qué pasaría si...? ¿Cómo afecta este tipo de problemas en la construcción de software?
- Explica y asegura comprensión del concepto de clase, su uso, definición, instanciación y diagrama.
- Entabla una conversación en base a las siguientes preguntas: ¿Cómo se puede representar y/o modelar un elemento tangible de la vida real en un concepto abstracto de software? ¿Cómo se puede generalizar un conjunto de objetos similares? ¿Cómo se puede identificar un objeto particular dentro la generalización de una clase?
- Entrega la guía **1.2_Guía de clases y objetos_Estudiante.docx**, para trabajar y guiar el aprendizaje en los desafíos a realizar.
- Presenta un diagrama de una Clase y entrega buenas prácticas para programación de clases.
- Presenta la **primera parte** del desafío **“Desafío 1: Abstracción - Creando clases”**: Observa a tu alrededor. Crea una sesión de Mentimeter (www.metimeter.com) para incluir las preguntas de **diapositiva 26** (la primera parte del desafío 1 solicita al docente esta acción), para apreciar la diversidad de respuestas de los estudiantes.
- Presenta la **segunda parte** del desafío **“Desafío 1: Abstracción - Creando clases”**: Creando clases, y luego conversa con sus estudiantes sobre los resultados de esta segunda parte del desafío.
- Invita a formar grupos de trabajo de tres integrantes. Presenta **“Desafío 2: Videojuego”** siendo la idea general el nuevo paradigma de programación orientado a objetos y plantea la pregunta ¿Es posible con esta nueva forma de programar, el desarrollo de videojuegos? Luego cumple con los siguientes pasos:

ETAPAS	DESCRIPCIÓN DE LAS ACTIVIDADES DE APRENDIZAJE
<h1 style="font-size: 48px; margin: 0;">2</h1> <p style="font-size: 24px; margin: 0;">Ejecución</p>	<ul style="list-style-type: none"> a. Idea general: El paradigma de la programación orientada a objetos. b. Pregunta esencial: ¿Es posible construir videojuegos usando esta forma de programar? c. Reto: Desarrollador de videojuego INDIE (ver diapositiva 30 en adelante). d. Preguntas, actividades y recursos guía: Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía de clases y objetos. e. Solución: Presenta las condiciones del entregable y consideraciones al momento de trabajar. f. Validación: Presenta los instrumentos de evaluación del “Desafío 2: Videojuego”, por medio del recurso 1.4_Instrumentos de evaluación de clases y objetos desafío 1 y 2_Estudiente.docx. g. Documentación y Publicación: Documenta y publica su trabajo en plataforma Github. <ul style="list-style-type: none"> • Insta a la autoevaluación de los desempeños a través del recurso 1.3_Instrumento de Autoevaluación_Estudiente.docx con preguntas de autorreflexión. <p>Actividad 2. Atributos y métodos</p> <ul style="list-style-type: none"> • Inicia la presentación 2.1_Presentación atributos y métodos.pptx para profundizar en conocimientos sobre atributos y métodos. Destaca y asegura la comprensión del concepto de atributos y métodos, mostrando ejemplos de las definiciones como por ejemplo “toString”. • Explica cómo se acceden a los métodos y atributos de una clase con el operador punto en las diapositivas 11 y 12. Invita a ver el video 1 explicativo que aborda el concepto de método. Conversa y profundiza sobre las buenas prácticas de programación para la creación de atributos y métodos. • Entrega guía 2.2_Guía de atributos y métodos_Estudiente.docx para reforzar contenidos abordados en la actividad y guiar aprendizajes en los desafíos a realizar. • Presenta primera parte del “Desafío 1: Criptomonedas” cumpliendo con los siguientes pasos: <ul style="list-style-type: none"> a. Idea general: Criptomonedas (Se sugiere a los estudiantes investigar en el enlace de la diapositiva 20 que son las criptomonedas). b. Pregunta esencial: ¿Cómo funcionan las transferencias de dinero en Internet? c. Reto: Crear una clase, siendo programadores de la criptomoneda CryptoPudú. (ver diapositiva 20 en adelante). d. Preguntas, actividades y recursos guía: Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre atributos y métodos. e. Solución: Presenta las condiciones del entregable y consideraciones al momento de trabajar. f. Validación: Presenta los instrumentos de evaluación del “Desafío 1: Criptomonedas”, por medio del recurso orientando aprendizaje por medio del recurso 2.4_Instrumento de evaluación de atributos y métodos desafíos 1 y 2_Estudiente.docx. g. Documentación y Publicación: Documenta y publica su trabajo en plataforma Github. • Una vez completada la actividad 8, presenta la segunda parte del “Desafío 1: Criptomonedas”, realizando un proceso de pruebas de código en el programa realizado previamente. • Presenta “Desafío 2: Tortugas” cumpliendo con los siguientes pasos: <ul style="list-style-type: none"> a. Idea general: Elementos gráficos en POO. b. Pregunta esencial: ¿Es posible dibujar en POO? c. Reto: Crear clase Tortuga (ver diapositiva 25 en adelante). Recordar a estudiantes que Repl.it es una herramienta Cloud para desarrollo de software. d. Preguntas, actividades y recursos guía: Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía atributos y métodos. e. Solución: Presenta las condiciones del entregable y consideraciones al momento de trabajar. f. Validación: Presenta los instrumentos de evaluación del “Desafío 2: Tortugas”, por medio del recurso orientando aprendizaje por medio del recurso 2.4_Instrumento de evaluación de atributos y métodos desafíos 1 y 2_Estudiente.docx. g. Documentación y Publicación: Documenta y publica su trabajo en plataforma Repl.it. • Insta a la autoevaluación de los desempeños a través del recurso 2.3_Instrumento de Autoevaluación_Estudiente.docx con preguntas de autorreflexión.

ETAPAS

DESCRIPCIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

2

Ejecución

Actividad 3. Parámetros y contexto estático

- Inicia la presentación **3.1_Presentación paso de parámetros y contexto estático_Docente.docx** para profundizar en la identificación de pasos de parámetros, haciendo referencia a dos tipos de datos y al paso por valor. Luego introduce el concepto de referencia y entabla conversación respecto a la pregunta de la **diapositiva 6**, que respondida en la **diapositiva 7**. Continúa con la presentación e invita a revisar el **video 1**, disponible en la **diapositiva 14**. Destaca y asegura la comprensión del concepto contexto estático, proporcionando ejemplos y observaciones de este concepto. Analizan el **video 2**, disponible en la **diapositiva 20**.
- Entrega y asegura comprensión de la guía **3.2_Guía de Paso de parámetros y contexto estático_Estudiante.docx** para reforzar aspectos abordados en la actividad y guiar aprendizajes de desafíos a realizar.
- Presenta la primera parte de **“Desafío 1: Healthsystem”** identificando:
 - a. **Idea general:** Mensajes en POO.
 - b. **Pregunta esencial:** ¿Cómo se comunican los objetos en POO?
 - c. **Reto:** Crear sistema de vida del personaje principal y zombis (**ver diapositiva 22 en adelante**).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía de parámetros y contexto estático.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del **“Desafío 1: Healthsystem”**, por medio del recurso **3.3 Instrumento de evaluación de paso de parámetros y contexto estatico Estudiante.docx**.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github.
- Una vez completada la actividad 8, presenta la segunda parte del **“Desafío 1: Healthsystem”**, realizando un proceso de pruebas de código en el programa realizado previamente.
- Presenta la primera parte de **“Desafío 2: GeoCalc”** identificando:
 - a. **Idea general:** Contexto estático.
 - b. **Pregunta esencial:** ¿Qué puedo hacer con los métodos estáticos?
 - c. **Reto:** Crear clase GeoCalc que permita calcular el área y perímetro de figuras geométricas (**ver diapositiva 28 en adelante**).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía de parámetros y contexto estático.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del **“Desafío 2: GeoCalc”**, por medio del recurso **3.3 Instrumento de evaluación de paso de parámetros y contexto estático_Estudiante.docx**.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github
- Una vez completada la actividad 8, presenta la segunda parte del **“Desafío 2: Geocalc”**, realizando un proceso de pruebas de código en el programa realizado previamente.
- Presenta la primera parte de **“Desafío 3: Estudiantes”** identificando:
 - a. **Idea general:** Contexto estático.
 - b. **Pregunta esencial:** ¿Qué puedo hacer con una variable estática?
 - c. **Reto:** Crear la clase Estudiante para modernizar el sistema de registro de nuevos estudiantes en una Universidad (**ver diapositiva 33 en adelante**).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía de parámetros y contexto estático.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del **“Desafío 3: Estudiantes”**, por medio del recurso **3.3 Instrumento de evaluación de paso de parámetros y contexto estatico_Estudiante.docx**.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Repl.it.
- Una vez completada la actividad 8, presenta la segunda parte del **“Desafío 3: Estudiantes”**, realizando un proceso de pruebas de código en el programa realizado previamente.

ETAPAS

DESCRIPCIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

2

Ejecución

Actividad 4. Control de acceso

- Inicia la presentación **4.1_Presentación control de acceso_–Docente.pptx** para introducir en conocimientos sobre el concepto de control de acceso y un ejemplo de un código que no posee esta característica. Presenta el **video 1** a modo de introducir a la temática. Entabla conversación sobre identificación de dos niveles de acceso para luego rehacer el ejemplo mencionado anteriormente y explica por qué es necesario implementar esta característica en POO.
- Introduce el concepto de scope o ámbito. Presenta la palabra reservada `this` y ejemplifica. Introduce los modificadores y accesores (`getters` y `setters`).
- Introduce el concepto de encapsulamiento. Analizan el **video 2**.
- Entrega y asegura comprensión de guía **4.2_Guía de control de acceso_Estudiente.docx** para reforzar contenidos y guiar aprendizajes en los desafíos a realizar.
- Presenta la primera parte de **“Desafío 1: Clínica de Michis”** identificando:
 - a. **Idea general:** Ciberseguridad.
 - b. **Pregunta esencial:** ¿Es posible hacer más segura mi aplicación?
 - c. **Reto:** Crear una solución tecnológica para modernizar la administración de clientes en una clínica de gatos (**ver diapositiva 19 en adelante**).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre control de acceso.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del **“Desafío 1: Clínica de Michis”**, por medio del recurso **4.3_Instrumento de evaluación de control de acceso_Estudiente.docx**.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github
- Una vez completada la actividad 8, presenta la segunda parte del **“Desafío 1: Clínica de Michis”**, realizando un proceso de pruebas de código en el programa realizado previamente.
- Presenta la primera parte de **“Desafío 2: GenPassword”** identificando:
 - a. **Idea general:** Ciberseguridad.
 - b. **Pregunta esencial:** ¿Por qué no es recomendable tener una contraseña como Pedrito123? Sugiere a los estudiantes revisar el enlace **“Contraseña Fuerte”** en la **diapositiva 24**.
 - c. **Reto:** Crear una clase de usuario con atributos de nombre y contraseña encapsulados (**ver diapositiva 24 en adelante**).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre control de acceso.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del **“Desafío 2: GenPassword”**, por medio del recurso **4.3_Instrumento de evaluación de control de acceso_Estudiente.docx**.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github
- Una vez completada la actividad 8, presenta la segunda parte del **“Desafío 2: GenPassword”**, realizando un proceso de pruebas de código en el programa realizado previamente.

Actividad 5. Paquetes y Java API

- Inicia la presentación **5.1_Presentación paquetes y java API–_Docente.pptx** para profundizar en conocimientos sobre paquetes y Java API. Da inicio a la actividad de paquetes y Java API, introduciendo el concepto de paquetes en Java. Asegura su comprensión.
- Menciona las acciones de importación y creación de paquetes en Java. Entrega un ejemplo de las acciones sobre paquetes. Responde a las preguntas ¿Por qué se usan los paquetes en Java? ¿Cuál es su relevancia?
- Introduce a través del **video 1** el concepto de API en Java. Asegura la comprensión de algunas estructuras de datos dinámicas provenientes de la API de Java. Evidencia la limitante de los arreglos al tener un tamaño fijo de datos. Asegura la comprensión de clase `ArrayList`, `Sets` y `Maps` mostrando ejemplos respectivos.
- Entrega guía **5.2_Guía de paquetes y java API_Estudiente.docx** para reforzar contenidos abordados en la actividad y guiar aprendizajes en los desafíos a realizar.

ETAPAS

DESCRIPCIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

2

Ejecución

- Presenta la primera parte de “**Desafío 1: Plataforma musical**” cumpliendo con los siguientes pasos:
 - a. **Idea general:** Aplicaciones cotidianas.
 - b. **Pregunta esencial:** ¿Cómo organizan los datos las grandes aplicaciones del mercado?
 - c. **Reto:** Crear una plataforma musical (**ver diapositiva 21 en adelante**).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre paquetes Java y API.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del “**Desafío 1: Criptomonedas**”, por medio del recurso **5.3_Instrumento de evaluación de paquetes y Java API_Estudiante.docx**.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github
- Una vez completada la actividad 8, presenta la segunda parte del “**Desafío 1: Plataforma musical**”, realizando un proceso de pruebas de código en el programa realizado previamente.
- Presenta la primera parte de “**Desafío 2: TechMap**” identificando:
 - a. **Idea general:** Industria 4.0.
 - b. **Pregunta esencial:** ¿Qué es la industria 4.0?
 - c. **Reto:** Crear programa para desarrollar un diccionario tecnológico (**ver diapositiva 25 en adelante**).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre paquetes Java y API.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del “**Desafío 2: TechMap**”, por medio del recurso **5.3_Instrumento de evaluación de paquetes y Java API_Estudiante.docx**.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github
- Una vez completada la actividad 8, presenta la segunda parte del “**Desafío 2: TechMap**”, realizando un proceso de pruebas de código en el programa realizado previamente.
- Presenta “**Desafío 3: java.presentation**” identificando:
 - a. **Idea general:** Reutilización de código.
 - b. **Pregunta esencial:** ¿Cómo los programadores utilizan la API de Java?
 - c. **Reto:** Investigar sobre diferentes paquetes usando la herramienta Random.org (**ver diapositiva 29 en adelante**).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre paquetes Java y API.
 - e. **Solución:** Presenta las condiciones de la presentación y consideraciones al momento de exponer lo investigado.
 - f. **Validación:** Presenta los instrumentos de evaluación del “**Desafío 3: java.presentation**”, por medio del recurso **5.3_Instrumento de evaluación de paquetes y Java API_Estudiante.docx**.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github

Actividad 6. Objetos compuestos e interfaz en la java

- Inicia la presentación **6.1_Presentación objetos compuestos e interfaz en java_Docente.pptx** para profundizar en conocimientos sobre objetos compuestos e interfaz en Java. Entabla una conversación a través de las siguientes preguntas ¿Qué pasa si un objeto se compone de otros objetos? ¿Qué objetos pueden ser compuestos? Presenta ejemplos de objetos compuestos en Java. Evidencia un diagrama de clases compuestas. Asegura comprensión e introduce el concepto de interfaz, sus características, métodos declarados (**analizan video 1**). Plantea la pregunta ¿Cuándo usar interfaces?
- Asegura comprensión e introduce el concepto de “**Casting**”.
Entrega guía **6.2_Guía de objetos compuestos e interfaz en java_Estudiante.docx** para reforzar contenidos abordados en la actividad y guiar aprendizajes en los desafíos a realizar.
- Presenta primera parte de “**Desafío 1: E-sports**”, cumpliendo con los siguientes pasos:
 - a. **Idea general:** Reutilización de código.
 - b. **Pregunta esencial:** ¿Cómo se estructura un sistema complejo en POO?
 - c. **Reto:** Crear clases de Coach y jugador para un portal de E-sports (**ver diapositiva 17 en adelante**).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre compuestos e interfaz en Java.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del “**Desafío 1: E-sports**”, por medio del recurso **6.3_Instrumento de evaluación de objetos compuestos e interfaz en Java_Estudiante.docx**.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github
- Una vez completada la actividad 8, presenta la segunda parte del “**Desafío 1: E-sports**”, realizando un proceso de pruebas de código en el programa realizado previamente.

ETAPAS

DESCRIPCIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

2

Ejecución

- Presenta primera parte de “Desafío 2: Analista deportivo” identificando:
 - a. **Idea general:** Ciencias de datos.
 - b. **Pregunta esencial:** ¿Cómo las grandes organizaciones utilizan los datos para su beneficio?
 - c. **Reto:** Crear clases como nadador, futbolista y una interfaz para una empresa de análisis deportivo (ver diapositiva 22 en adelante).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre compuestos e interfaz en Java.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del “Desafío 2: Analista deportivo”, por medio del recurso 6.3_Instrumento de evaluación de objetos compuestos e interfaz en Java_Estudiente.docx.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github
- Una vez completada la actividad 8, presenta la segunda parte del “Desafío 2: Analista deportivo”, realizando un proceso de pruebas de código en el programa realizado previamente.

Actividad 7. Herencia

- Inicia la presentación 7.1_Presentación herencia_Docente.pptx para profundizar en conocimientos sobre Herencia. Introduce con la actividad de herencia en Java, activando conocimientos con la pregunta ¿Qué significa heredar? y ¿Cuál es su relación con POO? Destaca y asegura comprensión de:
- Definición formal del concepto de herencia. Ejemplo de herencia, haciendo hincapié en la palabra extends y qué implica esto en el código. Pregunta: ¿Cómo se crea un constructor si los atributos se definen en otra clase? Da hincapié en la palabra reservada super.
- Da la posibilidad de crear herencia multinivel y entrega una referencia gráfica para entender el sistema creado. Presenta el video 1 para diversificar la construcción del aprendizaje.
- Entrega guía 7.2_Guía de herencia_Estudiente.docx para reforzar contenidos abordados en la actividad y guiar aprendizajes en el desafío a realizar.
- Presenta primera parte de “Desafío: Biblioteca Nacional”, cumpliendo con los siguientes pasos:
 - a. **Idea general:** Estructura.
 - b. **Pregunta esencial:** ¿Cómo podemos crear código limpio y ordenado?
 - c. **Reto:** Modernizar el sistema de libros y revistas del inventario de la biblioteca nacional (ver diapositiva 19).
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre Herencia.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación del “Desafío: Biblioteca Nacional”, mediante el recurso 7.3_Instrumento de evaluación de herencia_Estudiente.docx.
 - g. **Documentación y Publicación:** Documenta y publica su trabajo en plataforma Github
- Una vez completada la actividad 8, presenta la segunda parte del “Desafío: Biblioteca Nacional”, realizando un proceso de pruebas de código en el programa realizado previamente.

Actividad 8. Depuración y prueba de código

- Inicia la presentación 8.1_Presentación Depuración y pruebas de código_Docente.pptx para profundizar en conocimientos sobre depuración y pruebas de código en Java. En este sentido explica y asegura la comprensión del concepto de depurar o debug.
- Da inicio a una serie de pasos para detectar errores y aumentar la calidad del código realizado. El paso 1, llamado “No cometer errores”, realiza la importancia de crear código de buena calidad desde el comienzo, debido a que a medida que el proyecto se hace más grande, más complejo es corregir dichos errores. Menciona el concepto de patrones de diseño para recalcar que los programadores utilizan muchos recursos que ya están validados y que funcionan (ver recomendaciones de actividad 8).
- Menciona el concepto de diseño y la importancia del pseudocódigo. Presenta paso 2 llamado “Encuentra tus errores”, que busca detectar los errores potenciales y realizar pruebas de código para revisar la implementación de éste. Destaca que el programador debe tener noción de todos los caminos que genera su código. Denota la diferencia entre mayor, mayor igual, menor y menor igual. Presenta las herramientas de depuración más importantes que tienen los IDE de desarrollo (ver recomendaciones de actividad 8).

ETAPAS

DESCRIPCIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

2

Ejecución

- Asegura comprensión e introduce el concepto de prueba unitaria. Presenta el framework JUnit y un ejemplo de cómo tradicionalmente prueba código, para luego dar paso a la implementación de un Test realizado de manera correcta.
- Asegura la comprensión e introduce el concepto de Assertions, menciona los más importantes. Interactúa con sus estudiantes preguntando ¿Qué situación puede ser aplicado el assertEquals?, ¿assertNull? entre otros.
- Presenta el **paso 3** llamado **“Reproduce el error”** y denota la importancia de poder realizar esa acción ya que permite identificar la equivocación.
- Presenta el **paso 4**, llamado **“Genera una hipótesis”** que guarda relación con el error.
- Presenta el **paso 5** llamado **“Recopilar la información”**, que invita a los programadores a responder mediante datos la hipótesis generada en el paso anterior.
- Finaliza los pasos con **“Examinar la información”**, que invita a corroborar si la hipótesis es correcta o no y tomar las acciones descritas en la diapositiva.
- Destaca que esta serie de pasos guarda relevancia con el método científico y no solo es aplicable a la programación, si no a otros aspectos que tengan la misma naturaleza (arreglar un auto, jugar un partido de fútbol, etc).
- Refuerza algunos conceptos necesarios para la implementación de JUnit, que son las anotaciones.
- Asegura comprensión e introduce los conceptos de beforeClass y afterClass.
- Presenta la documentación oficial de JUnit.
- Asegura comprensión e introduce el concepto de excepciones.
- Asegura comprensión e introduce el concepto de manejo de excepciones y su gestión a través de cinco palabras claves.
- Asegura comprensión y ejemplifica el uso de las palabras clave (try, catch, finally, throw, throws) mostrando el **video 1**.
- Entrega guía **8.2_Guía de depuración y pruebas de código_Estudiente.docx** para reforzar contenidos abordados en la actividad y guiar aprendizajes en el desafío a realizar.
- Invita a formar grupos de cuatro personas y registra los equipos en la tabla (**diapositiva 54**). Presenta **“Desafío 1: CodeWars”** de la actividad, explicando el uso de la herramienta Codewars, desde la creación de una cuenta, el desarrollo de desafíos, la obtención de puntaje y la visualización del perfil de la cuenta. En este sentido, cumple con los siguientes pasos:
 - a. **Idea general:** Desarrollo guiado por pruebas.
 - b. **Pregunta esencial:** ¿Cómo las pruebas unitarias definen los límites de mis requerimientos?
 - c. **Reto:** Creación de una cuenta en CodeWars.
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre depuración y pruebas de código.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación y coevaluación del **“Desafío 1: CodeWars”**, por medio del recurso **8.3_Instrumento de evaluación de depuración y pruebas de código_Estudiente.docx**
 - g. **Documentación y Publicación:** Desarrolla un documento de texto con lo desarrollado por el equipo de trabajo.
 - h. Destaca que es una competencia entre los grupos y para ello deberán estar en constante comunicación para maximizar sus oportunidades de obtener un buen puesto. Posteriormente coevalúan sus desempeños a través de **8.4_Instrumento de coevaluación de módulo programación orientada a objetos_Estudiente.docx**.
- Presenta **“Desafío 2: Calidad del Software”** de la actividad, incorporando los elementos de testing, depuración y manejo de excepciones a desafíos de actividades anteriores.
 - a. **Idea general:** Calidad del Software.
 - b. **Pregunta esencial:** ¿Cómo construir un software de calidad?
 - c. **Reto:** Corrección de errores, implementación de JUnit y manejo de excepciones a los desafíos propuestos en la **Diapositiva 60**.
 - d. **Preguntas, actividades y recursos guía:** Preguntas generadas por sus estudiantes para el desarrollo exitoso del desafío presentado. Considerar la guía sobre depuración y pruebas de código.
 - e. **Solución:** Presenta las condiciones del entregable y consideraciones al momento de trabajar.
 - f. **Validación:** Presenta los instrumentos de evaluación y coevaluación del **“Desafío 2: Calidad del software”**, por medio del recurso **8.3_Instrumento de evaluación de depuración y pruebas de código_Estudiente.docx**
 - g. **Documentación y Publicación:** Desarrolla un documento de texto con lo desarrollado por el equipo de trabajo.

ETAPAS	DESCRIPCIÓN DE LAS ACTIVIDADES DE APRENDIZAJE
<p style="text-align: center;">2 Ejecución</p>	<p>Estudiantes</p> <ul style="list-style-type: none"> • Acuerdan y respetan las normas de convivencia. • Identifican, comprenden y activan los conocimientos previos, respondiendo preguntas gatilladoras. • Comprenden las explicaciones que realiza el/la docente respecto a los desafíos, evaluaciones y contenidos, participando activamente. • Toman apuntes y preguntan sobre los contenidos aplicados, y el uso de estos en contextos laborales reales. • Realizan los desafíos o retos asociados a cada actividad, guiando y orientando sus aprendizajes a través de los instrumentos de evaluación correspondientes. • Enlazan aprendizajes adquiridos con el proyecto a realizar en el módulo de Emprendimiento y empleabilidad. • Reciben resultados y retroalimentación del/la docente respecto a las evaluaciones.
<p style="text-align: center;">3 Cierre</p>	<p>Docente</p> <ul style="list-style-type: none"> • Realiza un proceso de síntesis de lo abordado en la clase. • Realiza las siguientes interrogantes: ¿Qué fue lo más difícil de las actividades? ¿Qué fue lo más fácil? ¿Creen que estos aprendizajes les serán útiles en su futuro? ¿Por qué? ¿Cuáles fueron los aprendizajes más significativos? ¿Por qué? ¿Qué aprendizajes faltaron por profundizar? • Retroalimenta en conceptos técnicos y teóricos sobre los elementos que se presentaron en la sección de ejecución. • Invita a participar activamente de la presentación de la siguiente actividad, para vincularlos a los contenidos de la siguiente clase. • Invita a participar del Discord de Programación TP. <p>Estudiantes</p> <ul style="list-style-type: none"> • Reflexionan sobre los desafíos ejecutados y su desempeño en cada uno de ellos. Destacan el para qué les ha servido efectuar las actividades, vinculando lo realizado en contextos laborales reales.

ESTRATEGIAS DE EVALUACIÓN

El módulo inicia con la aplicación de un diagnóstico a través del “Desafío 0”, con el propósito de identificar el nivel de conocimientos previos respecto a divisas.

En las actividades posteriores, se contempla como etapa inicial, dinámicas gatilladas por interrogantes que activen conocimientos previos y permitan compartir experiencias, intereses y motivaciones sobre los contenidos correspondientes en cada caso.

Se contempla realizar experiencias de aprendizaje situadas en la metodología activa Aprendizaje Basado en Retos, en la que se busca aplicar desafíos asociados a cada contenido. Para la evaluación de estos desafíos, se contemplan Rúbricas.

Se sugiere efectuar preguntas, profundizando en el para qué les ha servido realizar las actividades y su conexión con el contexto laboral real. Adicionalmente inste a generar espacios de autoevaluación y/o coevaluación de los desempeños en al menos las siguientes actividades 1, 2 y 8.

TIPOS DE INSTRUMENTOS DE EVALUACIÓN

Rúbrica

1.3_Instrumento de evaluación de clases y objetos desafío 0_Estudiante.docx

Rúbricas

1.4_Instrumentos de evaluación de clases y objetos desafío 1 y 2_Estudiante.docx

2.4_Instrumento de evaluación de atributos y métodos desafíos 1 y 2_Estudiante.docx

3.3_Instrumento de evaluación de paso de parámetros y contexto estatico_Estudiante.docx

4.3_Instrumento de evaluación de control de acceso_Estudiante.docx

5.3_Instrumento de evaluación de paquetes y Java API_Estudiante.docx

6.3_Instrumento de evaluación de objetos compuestos e interfaz en Java_Estudiante.docx

7.3_Instrumento de evaluación de herencia_Estudiante.docx

8.3_Instrumento de evaluación de depuración y pruebas de código_Estudiante.docx

Escalas de apreciación

1.5_Instrumentos de autoevaluación_Estudiante.docx

2.3_Instrumento de autoevaluación_Estudiante.docx

8.4_Instrumento de coevaluación de módulo programación orientada a objetos_Estudiante.docx

RECURSOS

Recursos generales aplicables a todas las actividades:

IDE de desarrollo: IntelliJ Idea, NetBeans, Visual Studio Code, Eclipse, entre otros.

Software de videoconferencia: Zoom, Meet, Teams, entre otros.

Noticias relacionadas con tecnología e industria 4.0:

Link 1: <https://www.genbeta.com/>

Link 2: <https://www.fayerwayer.com/>

Link 3: <https://www.tarreo.com/>

Link 4: <https://www.theverge.com/>

Link 5:

<https://gamedevelopment.tutsplus.com/tutorials/quick-tip-intro-to-object-oriented-programming-for-game-development--gamedev-1805>

Actividad 1. Clases y objetos

1.1_Presentación clases y objetos_Docente.pptx

Video 1:

https://www.youtube.com/watch?v=STVXkRO4LZY&feature=emb_logo

Video 2:

<https://www.youtube.com/watch?v=Nka4JSBgf7I&t>

1.2_Guía de clases y objetos_Estudiante.docx

1.3_Instrumento de evaluación de clases y objetos desafío 0_Estudiante.docx

1.4_Instrumentos de evaluación de clases y objetos desafío 1 y 2_Estudiante.docx

1.5_Instrumento de autoevaluación_Estudiante.docx

Actividad 2. Atributos y métodos

2.1_Presentación Atributos y métodos.pptx

Video 1:

https://www.youtube.com/watch?v=hDPSRbXB8U&feature=emb_title&ab_channel=Programaci%C3%B3nATS

Video 2:

https://www.youtube.com/watch?v=T5zkEpaMlfw&ab_channel=CristianRizzi

2.2_Guía de atributos y métodos_Estudiante.docx

2.3_Instrumento de evaluación de atributos y métodos desafíos 1 y 2_Estudiante.docx

2.4_Instrumento de autoevaluación_Estudiante.docx

RECURSOS

Actividad 3. Parámetros y contexto estático

3.1_Presentación Paso de parámetros y contexto estático_Docente.pptx

Video 1: <https://youtu.be/1MOjUdz0Y3Q>

Video 2: <https://youtu.be/aiBzXQmtzrA>

3.2_Guía de Paso de parámetros y contexto estático_Estudiante.docx

3.3_Instrumento de evaluación de paso de parámetros y contexto estático_Estudiante.docx

Actividad 4. Control de acceso

4.1_Presentación control de acceso_Docente.pptx

Video 1: <https://youtu.be/OKzN91ldizs>

Video 2: <https://youtu.be/8LjyCiuyNkw>

4.2_Guía de control de acceso_Estudiante.docx

4.3_Instrumento de evaluación de control de acceso_Estudiante.docx

Actividad 5. Paquetes y Java API

5.1_Presentación paquetes y Java API_Docente.pptx

Video 1: <https://youtu.be/0M6qHYdlqpc>

5.2_Guía de paquetes y Java API_Estudiante.docx

5.3_Instrumento de evaluación de paquetes y Java API_Estudiante.docx

Actividad 6. Objetos compuestos y e interfaz en la java

6.1_Presentación objetos compuestos e interfaz en Java_Estudiante.pptx

Video 1: <https://youtu.be/7MIB-K9AMxY>

6.2_Guía de objetos compuestos e interfaz en Java_Estudiante.docx

6.3_Instrumento de evaluación de objetos compuestos e interfaz en Java_Estudiante.docx

Actividad 7. Herencia

7.1_Presentación Herencia_Docente.pptx

Video 1: https://www.youtu.be/LZPafq9Ve_Q

7.2_Guía de Herencia_Estudiante.docx

7.3_Instrumento de evaluación de herencia_Estudiante.docx

Actividad 8. Depuración y pruebas de código

8.1_Presentación Depuración y pruebas de código_Docente.pptx

Video 1: <https://youtu.be/q9p1Zjplifo>

8.2_Guía de depuración y pruebas de código_Estudiante.docx

8.3_Instrumento de evaluación de depuración y pruebas de código_Estudiante.docx

8.4_Instrumento de coevaluación de módulo programación orientada a objetos_Estudiante.docx

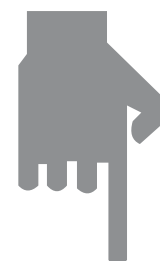
Se espera que estas actividades puedan contribuir a los procesos de enseñanza y aprendizaje, focalizadas en el desarrollo de competencias que fortalezcan el perfil de egreso de los y las estudiantes.

4. ANEXO 1. OTRAS ESTRATEGIAS DE EVALUACIÓN

PAUTA REFLEXIVA

¿EN QUÉ CONSISTE?

Es un tipo de evaluación formativa que se puede utilizar para comprobar rápidamente la comprensión de los estudiantes, ya que durante la clase se les da un momento de pausa para reflexionar sobre los conceptos e ideas que han sido enseñados o los procesos que han realizado para llegar a una respuesta. De esta forma, se espera que puedan realizar conexiones con los conocimientos previos, comentar algo que les pareció interesante y aclarar dudas.



¿CUÁNDO APLICARLA?

Durante la actividad

¿CÓMO APLICARLA?

Si bien puede ser usada en cualquier actividad, te recomendamos aplicar esta estrategia en actividades de comprensión. Pide a los estudiantes que se focalicen en las ideas clave del tema abordado en clase hasta ese momento y a través de preguntas, logra que reflexionen acerca de qué relaciones pueden establecer entre lo que están aprendiendo y sus conocimientos previos. Posteriormente pueden realizar preguntas aclaratorias, cuyas respuestas permitan obtener información para modificar a tiempo la enseñanza

EJEMPLOS



Se sugieren las siguientes preguntas:

- ¿Qué relaciones pueden establecer entre X y X?
- ¿A qué les recuerda lo que estamos estudiando?
- ¿Cómo podrían sintetizar esto?
- ¿Qué información podrían agregar?
- ¿Qué cosas aún no están claras?
- ¿Están teniendo alguna dificultad para establecer las relaciones?
- ¿Cómo podemos profundizar un poco más X idea?

RECOMENDACIÓN

Establecer previamente el tiempo de la pausa reflexiva, a razón de dos a tres minutos por pregunta.



4. ANEXO 1. OTRAS ESTRATEGIAS DE EVALUACIÓN

TICKET DE SALIDA

¿EN QUÉ CONSISTE?

Es una evaluación rápida e informal para el cierre de una clase, que permite registrar evidencias individuales al finalizar una actividad de aprendizaje mediante respuestas entregadas al salir de la sala.

¿CÓMO APLICARLA?

La mecánica de esta estrategia consiste en que durante los últimos dos o tres minutos de la clase, los estudiantes responden por escrito, una o más preguntas y la(s) entregan al salir de la sala, a modo de boleto o Ticket de salida.

Los tickets se van guardando de tal forma de poder generar un portafolio de evaluación para el estudiante.

¿PORQUÉ USARLA?

Permite recoger evidencias individuales sobre cómo los estudiantes han comprendido una actividad y además, implica un ejercicio metacognitivo, ya que les ayuda a reflexionar sobre lo que han aprendido y expresar qué o cómo están pensando la nueva información.



EJEMPLOS Y RECOMENDACIONES DE USO

Algunas opciones de Ticket de salida son:

- ¿Qué es lo más importante que aprendiste en la clase de hoy?
- ¿Cómo le explicarías X concepto a un amigo o amiga?
- ¿Entendiste la clase de hoy? ¿cómo lo sabes?
- ¿Qué preguntas tienes sobre la clase de hoy?
- ¿Qué puedo hacer para ayudarte?

Puede variar la forma de reflexión con opciones como escribir:

- Tres cosas que aprendieron.
- Dos dudas o preguntas que les quedaron.
- Una apreciación u opinión sobre la clase.

4. ANEXO 1. OTRAS ESTRATEGIAS DE EVALUACIÓN

ESCALERA DE RETROALIMENTACIÓN

Aplicación docente a estudiante durante la actividad

Daniel Wilson



Hacer sugerencias

Te sugiero...te ayudará a...



Expresar inquietudes

Me pregunto qué pasaría si cambias...
¿qué opinas?



Valorar

Destaco que hayas hecho...porque...sigue con esta práctica
en situaciones similares.



Aclarar

¿Qué querías decir con esto? ¿Por qué no incluiste x tema o
idea? ¿Cómo lo hiciste? ¿Qué te resultó fácil? ¿Qué dificultades
tuviste? ¿Cómo las superaste?

4. ANEXO 1. OTRAS ESTRATEGIAS DE EVALUACIÓN

ESCALERA DE METACOGNITIVA

Aplicación estudiante a sí mismo luego de la actividad

Robert Swartz



¿En qué otras ocasiones puedo usarlo?

¿Qué me ha resultado más fácil, más difícil,
más novedoso?

¿Cómo lo aprendí?

¿Qué aprendí?



ESPECIALIDAD PROGRAMACIÓN

MALETÍN DIDÁCTICO

PROGRAMACIÓN ORIENTADA A OBJETOS