



El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

CURSO VIRTUAL:

# Programación Básica **Java**



Duración:  
**7 SEMANAS**

**CICLO 2**

**‘Misión  
TIC 2022’**



# Acerca de este curso



Se realizarán 20 sesiones correspondientes al ciclo II de programación básica. En cada una de las sesiones se presentan los resultados de aprendizaje de cada ciclo (semana) y los específicos de cada sesión. Durante la sesión sincrónica interactiva, podrás participar y aplicar lo aprendido a través de un material base preparado para ti. Además, al final de cada sesión, se presentan los retos de práctica, que te ayudarán afianzar lo aprendido con el docente. Por último, de forma virtual tendrás a la mano: grabaciones, presentaciones, material de apoyo-complementario, apoyo de tutoría con monitores expertos y un reto que se establecerá con previo aviso para poder aplicar todo lo aprendido. Estos retos si serán calificables y tienen las siguientes características:

- Los retos tienen un contexto real.
- Los retos tienen un conjunto de requerimientos funcionales bien definidos.
- Los retos tienen un conjunto de pruebas asociados con cada uno de los requerimientos.
- Los retos son avisados previamente al beneficiario.
- Los retos son distintos para cada grupo de beneficiarios.
- Los retos son diseñados de manera que se puedan completar con los conocimientos y habilidades de las semanas anteriores y los que se van a ver en la semana actual.



# Resultados de aprendizaje

1. Desarrollar de manera individual un programa orientado por objetos en Java, monousuario, para resolver los requerimientos planteados por un tercero, que use una base de datos relacional, con una arquitectura MVC y con una interfaz gráfica.
2. Diseñar y utilizar una base de datos relacional simple usando SQL, y conectarla desde un programa usando JDBC.
3. Diseñar una aplicación utilizando objetos, expresando dicho diseño con un diagrama de clases de UML, con diagramas de casos de uso, y con un esquema básico de historias de usuario.
4. Diseñar y ejecutar un conjunto de pruebas unitarias para un programa.
5. Entender y utilizar los conceptos de interfaz y de herencia como mecanismos de reutilización.
6. Utilizar una arquitectura MVC para desacoplar la lógica de negocio de la interfaz de usuario.
7. Utilizar librerías y frameworks que apoyen el desarrollo de software.

El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

# Contenido del curso



SEMANA 1

## INTRODUCCIÓN A JAVA

SEMANA 2

## PROGRAMACIÓN ORIENTADA A OBJETOS

SEMANA 3

## PROGRAMACIÓN ORIENTADA A OBJETOS

SEMANA 4

## PROGRAMACIÓN ORIENTADA A OBJETOS

SEMANA 5

## APLICACIONES GRÁFICAS EN JAVA CONEXIÓN A BASE DE DATOS

SEMANA 6

## CONEXIÓN A BASE DE DATOS PRUEBAS UNITARIAS UML

SEMANA 7

MVC  
Proyecto IntegradorMisión  
TIC 2022

## ¿Qué aprenderás en este curso?

En este curso aprenderás a desarrollar de manera individual un programa orientado por objetos en Java, para resolver los requerimientos planteados por un tercero, teniendo en cuenta la arquitectura MVC y con una interfaz gráfica.

## Modalidad del curso

Las clases serán en las siguientes modalidades:

Clases sincrónicas interactivas. Espacios de trabajo donde profesor y estudiante interactúan en tiempo real durante el desarrollo de diferentes actividades por medio de herramientas TIC.

(Perveen, 2016, p. 22-23).

Duración: 2 horas y media. Tres veces por semana.

Nota: Toda sesión será grabada y estarán disponibles en la plataforma para que el beneficiario las pueda revisar con posterioridad.



Clases asincrónicas asistidas. Espacios de trabajo donde el estudiante desarrollará alguna actividad individual o colaborativa sin la interacción en tiempo real con el profesor, quien, a su vez, estará disponible monitoreando el desarrollo de las actividades durante el horario de clase establecido.

(Shahabadi and Uplane, 2015, p. 132).

Recordemos que el estudiante podrá tener apoyo de tutores en horario contrario a la sesión sincrónica, con una previa solicitud de cita, esto con el fin de complementar y guiador del proceso la solución del reto.



La nota final del ciclo 1 será individual y se calcula así:

- Reto 1 (10%)
- Reto 2 (10%)
- Reto 3 (20%)
- Reto 4 (20%)
- Reto 5 (20%)
- Inglés (20%)



Todo ejercicio práctico, revisión de material y actividades formativas del docente, se consideran evaluación formativa, que te ayudará al afianzamiento de tus habilidades y a los fundamentos de programación.

Sesión Sincrónica	Duración clase: 2:30 horas	
Tiempo estimado	Momento	Descripción actividad

Los estudiantes estarán en capacidad de:

- Determinar las componentes mínimas de un programa escrito en Java.
- Identificar y comprender los potenciales errores iniciales de un programa escrito en Java.
- Describir y aplicar el método print y println.
- Diferenciar Tipos de datos
- Declarar variables
- Manejo de Operadores aritméticos, relacionales y lógicos
- Precedencia de operadores
- Expresiones
- Asignación
- Manipular Tipos de datos Strings y Character.
- Convertir números a strings y de strings a números para entrada.
- Describir y aplicar el método Scanner.
- Describir los operadores relacionales
- Diseñar y desarrollar programas que incluyan condicionales If-else, if-else if-else simples y anidados.
- Diseñar y desarrollar programas que incluyan el manejo de switch.
- Diseñar y desarrollar programas que incluyan ciclo repetitivo while
- Diseñar y desarrollar programas que incluyan ciclo repetitivo do-while
- Operadores de asignación compuesto (operador=)
- Operadores de incremento y decremento(++ ,--)
- Diseñar y desarrollar programas que incluyan ciclo repetitivo for

15"

60"

Introducción

Componente Teórico

- Revisión de los objetivos del curso, la modalidad de trabajo y el tiempo estimado.

- Revisión de la plataforma para acceder al contenido del curso.
- Reglas.
- Instalación de las especificaciones de java en VSCode.
- Descripción de componentes principales y fijos de un programa

Java en entorno consola.

- Muestra del primer programa “Hola Mundo”
- Descripción del método de impresión System.out.print y System.out.println.
- Descripción del paso de argumentos en args.
- Muestra del segundo programa con “Hola Mundo <<Nombre-persona>>”, con el valor Nombre-persona pasado como argumento.
- Muestra de posibles errores que se cometen en la programación: falta de punto y coma, no abrir o cerrar un bloque de instrucciones con llaves ({,}),
- Revisión de Tipos de datos incorporado en Java.
- Estudio de Variables, Declaración de variables.
- Estudio de Operadores, Precedencia y Expresiones
- Asignación
- Estudio de Tipos de datos Strings y Character

Sesión Sincrónica

Duración clase: 2:30 horas

Tiempo estimado	Momento	Descripción actividad
60”		
60”		
15”		

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión de Conversión de números a Strings y de Strings a números para entrada.
- Estudiar y aplicar el método Scanner para la entrada de datos interactivamente.
- Estructuras condicionales simples y anidadas(if-else, if-else if-else, switch, while, do while, for.
- Revisión y estudio de los operadores de asignación compuesto y de incremento y decremento.
- Solución de problemas aplicando las estructuras algorítmicas y su sintaxis.
- Codificación en Java (Editor VSCode) de las estructuras algorítmicas a través de ejemplo con cada una (6 ejemplos) y un ejemplo que integre while con if-else y for con if-else.
- Retos para desarrollar en Java (Editor VS Code).
- Presentación de los retos para el resto del día.

RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.

Sesión Sincrónica

Duración clase: 2:30 horas

Tiempo estimado	Momento	Descripción actividad
-----------------	---------	-----------------------

Los estudiantes estarán en capacidad de:

- Declarar e inicializar arreglos (vectores).
- Utilizar los arreglos para almacenar datos y recuperarlos.
- Iterar a través de los arreglos y manipular sus datos utilizando las primitivas de programación estudiadas.

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

## RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Especificación de las dificultades encontradas.
- Repaso y aplicación de arreglos en la programación básica.
- Revisión de la estructura sintáctica de declaración e inicialización de arreglos unidimensionales.
- Desarrollar ejercicios y programas donde se muestre la forma de almacenar y recuperar información de los arreglos.
- Desarrollar aplicaciones de los arreglos que incluyan iteraciones en sus recorridos, y aplicando las primitivas de programación previamente estudiadas.
- Solución de problemas aplicando arreglos.
- Codificación en Java (Editor VS Code) de las aplicaciones de arreglos unidimensionales.
- Retos para desarrollar en Java (Editor VS Code).
- Presentación de los retos para el resto del día.

## RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.



Tema

INTRODUCCIÓN A JAVA

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

# Sesión Sincrónica

Los estudiantes estarán en capacidad de:

- Declarar e inicializar arreglos (matrices).
- Utilizar los arreglos para almacenar datos y recuperarlos.
- Iterar a través de los arreglos y manipular sus datos utilizando las primitivas de programación estudiadas.

**Duración clase: 2:30 horas**

Tiempo estimado	Momento	Descripción actividad
15"		
60"		
60"		
15"		
Introducción		
Componente Teórico		
Componente Práctico		
Revisión del Reto		

**Actividad trabajo independiente**

## RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Especificación de las dificultades encontradas en el desarrollo del reto del día anterior sobre el uso de arreglos
- Repaso y aplicación de arreglos en Java según lo comprendido el día anterior.
- Desarrollar aplicaciones de los arreglos bidimensionales que incluyan iteraciones en sus recorridos, y aplicando las primitivas de programación previamente estudiadas.
- Solución de problemas prácticos y del contexto real aplicando
- arreglos.
- Codificación en Java (Editor VS Code) de las aplicaciones de arreglos bidimensionales.
- Retos para desarrollar en Java (Editor VS Code).
- Presentación de los retos para el resto del día.

## RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.

Vigilada Mineducación

Los estudiantes estarán en capacidad de:

- Manejar la terminología básica de la POO a partir de una analogía con elemento de la vida cotidiana.
- Definir una clase a partir de este elemento.
- Definir los métodos y atributos de la clase creada a partir del elemento.
- Instanciar una clase para crear un objeto.
- Declarar una clase con un método sin paso de parámetros.
- Instanciar el objeto a partir de la clase.

**Duración clase: 2:30 horas**

Momento

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

- Revisión del reto del día anterior.
- Especificación de las dificultades encontradas en el desarrollo del reto del día anterior sobre el uso de arreglos.
- Establecer un ejemplo con un elemento (auto, perro, árbol, camisa, etc) de la vida cotidiana que sirva de analogía para introducir el concepto de objeto.
- Continuar con la situación de la vida cotidiana y definir los componentes que definen al elemento seleccionado como un objeto.
- Establecer dentro del elemento cuáles son las tareas que ejecuta (métodos) y cuáles son sus características propias (atributos).
- Generalizar los conceptos de clases, métodos, atributos, instancias, y objetos.
- Definir y estructurar el programa que incluirá una clase y un método sin paso de parámetros
- Definir el programa que llevará a cabo la instanciación de la clase anterior.
- Construir y probar en Java (Editor VS Code) el programa con la clase definida previamente y el programa desde el cual se lleve a cabo la instanciación
- Reto del día en forma de quiz para evaluar una situación de analogía similar a la presentada en la clase y donde respondan los conceptos desarrollados.

Los estudiantes estarán en capacidad de:

- Definir el diagrama de clase para la clase creada con método sin paso de parámetros.
- Declarar un método con paso de parámetros.
- Definir el diagrama de clase para la clase creada con método sin paso de parámetros.
- Implementar los atributos de una clase como variables de instancias.
- Definir el diagrama de clase aplicando variables de instancias y el uso de los métodos set y get.

**Duración clase: 2:30 horas**

Momento

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Los estudiantes estarán en capacidad de:

- Especificar cómo se inicializa un objeto cuando no existe un constructor.
- Inicializar objetos utilizando un constructor.
- Definir el diagrama de clase insertando el constructor.
- Mostrar la existencia de los paquetes del API de Java y explicar los más importantes.
- Explicar los conceptos asociados a metodos y campos static.

**Duración clase: 2:30 horas**



Momento

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Sesión Sincrónica	Duración clase: 2:30 horas	
Tiempo estimado	Momento	Descripción actividad

Los estudiantes estarán en capacidad de:

- Aplicar adecuamente el cast de un tipo de dato a otro y de igual forma usar la promoción de argumentos en un método.
- Usar la referencia this y demostrar su uso en un programa de aplicación
- Explicar y aplicar el encapsulamiento en Java de variables y métodos a través de la construcción de una aplicación
- Definir y aplicar los métodos set y get con variables de instancias
- Implementar los métodos set y get para modificar las variables privadas definidas dentro del encapsulamiento.

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Demostrar el uso del cast y la promoción de argumentos en un método con aplicaciones.
- Explicar la referencia this y demostrar con un programa su uso.
- Explicar con una programa el uso del encapsulamiento de variables y métodos haciendo uso de los modificadores private y public de distintas formas para mostrar esta característica de la POO.
- Implementar los métodos set y get para acceder a las variables privadas definidas para el encapsulamiento.
- Definir y estructurar el programa que incluirá una clase con el uso de los métodos set y get.
- Retos para desarrollar en Java (Editor VS Code).
- Presentación de los retos para el resto del día.

## RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Sesión Sincrónica	Duración clase: 2:30 horas	
Tiempo estimado	Momento	Descripción actividad

Los estudiantes estarán en capacidad de:

- Reconocer el uso permitido y no permitido de los métodos y variables con modificadores de acceso public y private.
- Definir y construir un programa que demuestre los usos de los modificadores public y private.
- Explicar la declaración de main como static y aplicarlo en un contexto práctico.
- Explicar la aplicación conjunta de las palabras claves public y static en un método.

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

## RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Explicación de los modificadores de acceso public y private
- Explicar la naturaleza static de main y mostrar un ejemplo
- Explicar la aplicacion conjunta de las palabras claves public y static en un método y mostrarla en un programa ejemplo.
- Demostrar con un programa el uso restringido de los modifica- dores public y private.
- Retos para desarrollar en Java (Editor VS Code).
- Presentación de los retos para el resto del día.

## RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Sesión Sincrónica	Duración clase: 2:30 horas	
Tiempo estimado	Momento	Descripción actividad

Los estudiantes estarán en capacidad de:

- Definir y aplicar los conceptos relacionados con herencia en POO.
- Especificar los contextos donde se puede aplicar la herencia.
- Diseñar la herencia entre clases: definición de superclase y subclases.
- Explicar el uso del tipo de acceso protected y su aplicación en herencia.
- Construir programas en Java con la aplicación de los conceptos básicos de herencia.

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Explicación de los conceptos de herencia.
- Exponer los contextos de aplicación de herencia.
- Explicar el diseño de la herencia entre clases: definición de superclase y subclases a través de ""..es un.."".
- Definición de términos: clase padre o base o superclase y clase hija o derivada o subclase.
- Explicación del tipo de acceso protected.
- Desarrollar y ejecutar programas de aplicación que muestre los conceptos de clase principal (padre) y clases hijas.
- Retos para desarrollar en Java (Editor VS Code).
- Presentación de los retos para el resto del día.

RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Sesión Sincrónica

Duración clase: 2:30 horas

Tiempo estimado

Momento

Descripción actividad

Los estudiantes estarán en capacidad de:

- Explicar el desarrollo de una clase padre con varios constructores con parámetros y el constructor por defecto.
- Explicar el desarrollo de un constructor a partir de la clase padre con super sin paso de parámetros y con paso de parámetros.
- Construir programas en Java con la aplicación de los conceptos de herencia aplicando el método super.
- Sobre-escribir métodos heredados en la clase hija utilizando el método super.

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

## RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Exponer el desarrollo de una clase padre con varios constructores con parámetros y el constructor por defecto.
- Explicar el desarrollo de un constructor a partir de la clase padre con super sin paso de parámetros y con paso de parámetros.
- Explicar cuándo sobre-escribir métodos heredados por uno nuevo dentro de la clase hija utilizando el método super.
- Explicar con el desarrollo de programas en Java la aplicación de los conceptos de herencia aplicando el método super en distintas modalidades.
- Retos para desarrollar en Java (Editor VS Code).
- Presentación de los retos para el resto del día.

## RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Sesión Sincrónica

Duración clase: 2:30 horas

Tiempo estimado

Momento

Descripción actividad

Los estudiantes estarán en capacidad de:

- Explicar el polimorfismo básico y su relación con la herencia en Java .
- Explicar el polimorfismo utilizando los conceptos de clase abstracta y la sobreescritura de métodos (@Override).
- Construir programas que usen el polimorfismo básico con el apoyo de la herencia y con el uso de la clase abstracta y la sobreescritura de métodos.

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

## RETO DEL DÍA:



- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.
- Revisión del reto del día anterior.
  - Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
  - Explicar los conceptos de polimorfismo
  - Aplicar los conceptos de polimorfismo básico a partir de los resultados obtenidos de la herencia utilizando el mismo programa desarrollado en la clase previa. - Mostrar distintas situaciones donde se exponga claramente el concepto de polimorfismo sin utilizar los conceptos cd Override y abstract.
  - Explicar los conceptos de clase abstracta y la sobreescritura.
  - Aplicar los conceptos de polimorfismo utilizando la clase abstracta y la sobreescritura en un programa.
  - Mostrar distintas situaciones donde se exponga claramente el concepto de polimorfismo utilizando los conceptos Override y abstract.
  - Definir y estructurar un programa donde se exponga claramente el concepto de polimorfismo utilizando los conceptos OverrEditor y abstract
  - Retos para desarrollar en Java (Editor VS Code).
  - Presentación de los retos para el resto del día.

RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Los estudiantes estarán en capacidad de:

- Explicar los conceptos de Interfaz para manejar herencias múltiples
- Desarrollar un progrma donde aplique los conceptos de interfaz en conjunto con el concepto de herencia de una clase.

Duración clase: 2:30 horas

Momento

Descripción actividad

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

RETO DEL DÍA:  
APLICACIONES GRÁFICAS EN JAVA

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Los estudiantes estarán en capacidad de:

- Usar el Editor (VS Code) para el desarrollo de aplicaciones en Java.
- Desarrollar interfaces gráficas utilizando codificación directa.

Sesión Sincrónica

Duración clase: 2:30 horas

Tiempo estimado

Momento

Descripción actividad

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Explicación breve de los conceptos asociados a la codificación de interfaz gráfica y sus métodos.
- Desarrollo de una aplicación usando codificación directa para crear una ventana desde VS Code.
- Retos para desarrollar en Java (Editor VSCode).
- Presentación de los retos para el resto del día.

RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.

APLICACIONES GRÁFICAS EN JAVA

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Los estudiantes estarán en capacidad de:

- Desarrollar interfaces gráficas en VS Code.

Duración clase: 2:30 horas

Momento

Descripción actividad

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

RETO DEL DÍA:

- Retos para desarrollar en Java (Editor VS Code).
- Presentación de los retos para el resto del día.

# CONEXIÓN A BASE DE DATOS

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Los estudiantes estarán en capacidad de:

- Definir y diseñar una base de datos relacional
- Construir una base de datos en SQLite de una tabla
- Manipular la gestión de información en la base de datos construida
- Explicar y aplicar el concepto de conexión a una base de datos relacional (JDBC)
- Construir una aplicación con entorno gráfico que conecte a una base de datos relacional y lleve a cabo operaciones sobre esta.

Duración clase: 2:30 horas

Momento	Descripción actividad
---------	-----------------------

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

## RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Explicar los conceptos de una base de datos relacional, su diseño y operaciones.
- Explicación del entorno SQLite y la forma en que se define una base de datos en éste.
- Diseñar y construir una base de datos en SQLite de una
- tabla.
- Demostar el uso de la base de datos construida.
- Explicar mediante la construcción de una aplicación con entorno gráfico la conexión a la base de datos relacional (JDBC) y la ejecución de operaciones sobre esta.
- Retos para desarrollar en el Editor de VS Code usando SQLite.
- Presentación de los retos para el resto del día.

## RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos.

# CONEXIÓN A BASE DE DATOS - PRUEBAS UNITARIAS

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Sesión Sincrónica

Duración clase: 2:30 horas

Tiempo estimado

Momento

Descripción actividad

Los estudiantes estarán en capacidad de:

- Definir y diseñar una base de datos relacional de más de una tabla con relaciones
- Construir una base de datos en SQLite con varias relaciones
- Manipular la gestión de información en la base de datos construida
- Construir una aplicación constituida de varios métodos utilizando entorno gráfico que conecte a base de datos relacional y lleve a cabo operaciones sobre esta
- Explicar los conceptos de pruebas de pruebas unitarias.

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

## RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Explicar el concepto de conexión a una base de datos relacional desde Java (JDBC)
- Explicar los conceptos de relaciones entre tablas
- Demostar el uso de la base de datos construida.
- Definición de pruebas unitarias
- Diseñar y construir una base de datos en SQLite con varias tablas (no más de 3) y sus relaciones.
- Explicar mediante la construcción de una aplicación con varios métodos dentro de un entorno gráfico la conexión a la base de datos relacional y la ejecución de operaciones sobre esta.
- Retos para desarrollar en el Editor de VS Code usando SQLite.
- Presentación de los retos para el resto del día.

## RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos.

## CONEXIÓN A BASE DE DATOS - PRUEBAS UNITARIAS

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Los estudiantes estarán en capacidad de:

- Manipular la base de datos relacional de más de una tabla con relaciones construida en SQLite.
- Desarrollar una aplicación constituida de varios métodos utilizando entorno gráfico que conecte a base de datos relacional y lleve a cabo operaciones sobre esta.
- Definir y aplicar pruebas unitarias sobre los distintos métodos desarrollados en una aplicación.



Duración clase: 2:30 horas

Momento	Descripción actividad
15"	
60"	
60"	
15"	
Introducción	
Componente Teórico	
Componente Práctico	
Revisión del Reto	

Actividad trabajo independiente

RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Continuar con la construcción de los métodos de la aplicación con entorno gráfico y conexión a la base de datos relacional y la ejecución de operaciones sobre esta.
- Definir y llevar a cabo las pruebas unitarias sobre los métodos.
- Retos para desarrollar en el Editor de VS Code usando SQLite.
- Presentación de los retos para el resto del día.

RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos.

UML

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Sesión Sincrónica

Duración clase: 2:30 horas

Tiempo estimado	Momento	Descripción actividad
-----------------	---------	-----------------------

Los estudiantes estarán en capacidad de:

- Definir UML y su importancia.
- Identificar y aplicar los principales diagramas de UML.
- Aplicar los diagramas de UML (diagramas de casos de uso, de clases, relaciones entre diagramas de clases) en el diseño de una aplicación a partir de los requerimientos de un usuario.

15"  
60"  
60"  
15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

## RETO DEL DÍA:

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Explicar los conceptos de UML y su importancia.
- Definir los diagramas de UML más relevante: casos de uso, diagramas de clases, relación entre diagramas de clases.
- Explicar los diagramas UML sobre la aplicación contstruida con bases de datos y el entorno gráfico en clases previas a partir de los requerimientos del usuario.
- Aplicar los diagramas UML con otros requerimientos de usuario que deben ser completados como parte del reto diario.
- Reto del día en forma de quiz para evaluar la aplicación de los conceptos y diagramas de UML a partir de los requerimientos dados.

## RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.

## RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos. y se carga la nota en las calificaciones del curso.

## MVC

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Sesión Sincrónica	Duración clase: 2:30 horas	
Tiempo estimado	Momento	Descripción actividad

Los estudiantes estarán en capacidad de:

- Explicar y aplicar el Modelo Vista Controlador a partir de los requerimientos generados por un tercero.
- Aplicar los conceptos de UML para definir los diagramas correspondientes.
- Definir la base de datos relacional que comprende la solución del requerimiento.
- Construir una aplicación usando un entorno gráfico y con conexión a la base de datos relacional bajo el concepto de MVC utilizando los conceptos de POO.
- Llevar a cabo pruebas unitarias sobre los métodos construidos.

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

Actividad trabajo independiente

## RETO DEL DÍA:

- Se presentan retos diarios e individuales asociados a los aprendizajes esperados en la sesión. Los resultados son sometidos a una valoración automática en la cuál se

presenta una retroalimentación de los resultados.

- Al inicio de cada sesión se discuten los retos presentados en el día anterior como espacio de reflexión y metacognición de las pruebas.
- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Explicar el concepto de Modelo Vista Controlador.
- Explicar el proyecto integrador que conjugue todos los conceptos desarrollados: POO, Bases de datos, UML, y Pruebas unitarias.
- Definir los módulos que desarrollarán en cada sesión de trabajo.
- Acompañar en el desarrollo de los módulos iniciales del proyecto integrador.
- Retos para desarrollar en el Editor de VS Code usando SQLite.
- Presentación de los retos para el resto del día.

RETO SEMANAL:

- Se propone un reto semanal e individual que contenga todos los aprendizajes que se abordaron a lo largo de la semana. Este reto será calificado de forma automática (Codegrade) entregando una retroalimentación al estudiante de los resultados obtenidos.

MVC PROYECTO INTEGRADOR

RESULTADO DE APRENDIZAJE DE LA SESIÓN:

Sesión Sincrónica	Duración clase: 2:30 horas	
Tiempo estimado	Momento	Descripción actividad

Los estudiantes estarán en capacidad de:

- Explicar y aplicar el Modelo Vista Controlador a partir de los requerimientos generados por un tercero.
- Aplicar los conceptos de UML para definir los diagramas correspondientes.
- Definir la base de datos relacional que comprende la solución del requerimiento.
- Construir una aplicación usando un entorno gráfico y con conexión a la base de datos relacional bajo el concepto de MVC utilizando los conceptos de POO.
- Llevar a cabo pruebas unitarias sobre los métodos contruidos.

15"

60"

60"

15"

Introducción

Componente Teórico

Componente Práctico

Revisión del Reto

RETO DEL DÍA:

- Revisión del reto del día anterior.
- Recordar los conceptos previos a través de preguntas para aplicarlos en el desarrollo del tema del día.
- Acompañar en el desarrollo de los módulos del proyecto integrador.
- Acompañar en el desarrollo de los módulos del proyecto integrador
- Retos para desarrollar en el Editor de VS Code usando SQLite.
- Presentación de los retos para el resto del día.



**¡LA CASA  
QUE ENSEÑA  
A TRIUNFAR!**

**‘Mision  
TIC 2022’**