



UES

Universidad Estatal de Sonora
La Fuerza del Saber Estimulará mi Espíritu

MANUAL DE PRÁCTICAS DE LABORATORIO

Programacion Estructurada Laboratorio

Programa Académico
INGENIERIA ENSOFTWARE
Fecha de elaboración
03/07/2025
Versión del Documento
1era. Versión



Dra. Martha Patricia Patiño Fierro
Rectora

Mtra. Ana Lisette Valenzuela Molina
**Encargada del Despacho de la Secretaría
General Académica**

Mtro. José Antonio Romero Montaña
Secretario General Administrativo

Lic. Jorge Omar Herrera Gutiérrez
**Encargado de Despacho de Secretario
General de Planeación**

Tabla de contenido

INTRODUCCIÓN.....	4
IDENTIFICACIÓN	5
<i>Carga Horaria del alumno</i>	<i>5</i>
<i>Consignación del Documento</i>	<i>5</i>
MATRIZ DE CORRESPONDENCIA	6
NORMAS DE SEGURIDAD Y BUENAS PRÁCTICAS.....	7
<i>Reglamento general del laboratorio</i>	<i>7</i>
<i>Reglamento de uniforme.....</i>	<i>7</i>
<i>Uso adecuado del equipo y materiales.....</i>	<i>8</i>
<i>Manejo y disposición de residuos peligrosos.....</i>	<i>8</i>
<i>Procedimientos en caso de emergencia</i>	<i>8</i>
RELACIÓN DE PRÁCTICAS DE LABORATORIO POR ELEMENTO DE COMPETENCIA....	8
PRÁCTICAS.....	12
FUENTES DE INFORMACIÓN	31
NORMAS TÉCNICAS APLICABLES.....	32
ANEXOS	3

INTRODUCCIÓN

Como parte de las herramientas esenciales para la formación académica de los estudiantes de la Universidad Estatal de Sonora, se definen manuales de práctica de laboratorio como elemento en el cual se define la estructura normativa de cada práctica y/o laboratorio, además de representar una guía para la aplicación práctica del conocimiento y el desarrollo de las competencias clave en su área de estudio. Su diseño se encuentra alineado con el modelo educativo institucional, el cual privilegia el aprendizaje basado en competencias, el aprendizaje activo y la conexión con escenarios reales.

Con el propósito de fortalecer la autonomía de los estudiantes, su pensamiento crítico y sus habilidades para la resolución de problemas, las prácticas de laboratorio integran estrategias didácticas como el aprendizaje basado en proyectos, el trabajo colaborativo, la experimentación guiada y el uso de tecnologías educativas. De esta manera, se promueve un proceso de enseñanza-aprendizaje dinámico, en el que los estudiantes no solo adquieren conocimientos teóricos, sino que también desarrollan habilidades prácticas y reflexivas para su desempeño profesional.

Señalar en este apartado brevemente los siguientes elementos según corresponda:

- Propósito del manual
- Justificación de su uso en el programa académico
- Competencias a desarrollar
 - **Competencias blandas:** Habilidades transversales que se refuerzan en las prácticas, como la comunicación, el trabajo en equipo, el uso de tecnologías, etc.
 - **Competencias disciplinares:** Conocimientos específicos del área del laboratorio, incluyendo fundamentos teóricos y habilidades técnicas.
 - **Competencias profesionales:** Aplicación de los conocimientos adquiridos en escenarios reales o simulados, en concordancia con el perfil de egreso del programa.

IDENTIFICACIÓN

Nombre de la Asignatura		programación Estructurada	
Clave	061CP040	Créditos	6
Asignaturas Antecedentes		Plan de Estudios	Ingeniería en Software

Área de Competencia	Competencia del curso
Desarrollar software y servicios de soporte técnico y redes, con la finalidad de solucionar problemas y agilizar procesos en la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional, a través del análisis de problemas, comunicación, liderazgo e innovación.	Desarrollar programas mediante el uso de lenguajes con un enfoque estructurado, en un entorno de desarrollo integrado, con el fin de dar solución a los problemas y servir de apoyo en la toma de decisiones dentro de las organizaciones, aplicando los principios de una buena programación poniendo en práctica el análisis de problemas, el trabajo en equipo y la responsabilidad.

Carga Horaria de la asignatura

Horas Supervisadas			Horas Independientes	Total de Horas
Aula	Laboratorio	Plataforma		
3	1	1		5

Consignación del Documento

Unidad Académica	Unidad Académica Hermosillo
Fecha de elaboración	28/06/2025
Responsables del diseño	Margarita Soto Rodriguez
Validación	
Recepción	Coordinación de Procesos Educativos

MATRIZ DE CORRESPONDENCIA

Señalar la relación de cada práctica con las competencias del perfil de egreso

PRÁCTICA	PERFIL DE EGRESO
<p>Práctica No. 1 Programas en Java para entrada y salida de datos por consola.</p>	<p>Desarrollar software con la finalidad de agilizar los procesos y la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional con enfoque de liderazgo</p>
<p>Práctica No. 2 Programas en Java aplicando la estructura selectiva if...else.</p>	<p>Desarrollar software con la finalidad de agilizar los procesos y la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional con enfoque de liderazgo</p>
<p>Práctica No. 3 Práctica de programación en Java aplicando la estructura selectiva switch.</p>	<p>Desarrollar software con la finalidad de agilizar los procesos y la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional con enfoque de liderazgo</p>
<p>Práctica No. 4 Programas en Java aplicando la estructura repetitiva for.</p>	<p>Desarrollar software con la finalidad de agilizar los procesos y la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional con enfoque de liderazgo</p>
<p>Práctica No. 5 Ejercicios en Java aplicando la estructura repetitiva while</p>	<p>Desarrollar software con la finalidad de agilizar los procesos y la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional con enfoque de liderazgo</p>
<p>Práctica No. 6 Práctica de programación en Java aplicando la estructura repetitiva do while</p>	<p>Desarrollar software con la finalidad de agilizar los procesos y la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional con enfoque de liderazgo</p>

<p>Práctica No. 7 Ejercicios en Java usando arreglos unidimensionales</p>	<p>Desarrollar software con la finalidad de agilizar los procesos y la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional con enfoque de liderazgo</p>
<p>Práctica No.8 Programas en Java usando arreglos bidimensionales</p>	<p>Desarrollar software con la finalidad de agilizar los procesos y la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional con enfoque de liderazgo</p>
<p>Práctica No.9 Programas en Java usando métodos de clase o estáticos.</p>	<p>Desarrollar software con la finalidad de agilizar los procesos y la toma de decisiones en empresas públicas y privadas, bajo estándares de calidad nacional e internacional con enfoque de liderazgo</p>

NORMAS DE SEGURIDAD Y BUENAS PRÁCTICAS

Reglamento general del laboratorio

- Se debe ingresar al laboratorio puntualmente según el horario asignado.
- La asistencia será registrada por el docente o responsable técnico.
- Comportamiento responsable y respetuoso
- Mantener el orden, respeto y silencio dentro del laboratorio.
- No se permite consumir alimentos ni bebidas.
- Los dispositivos móviles deben estar en modo silencioso o vibrador y su uso solo se permitirá con fines académicos.
- Cumplimiento académico
- Cada alumno debe seguir las instrucciones del docente y cumplir con las actividades asignadas dentro del tiempo establecido.
- La práctica debe realizarse de manera individual o en equipo, según se indique.
- Salud y ergonomía
- Usar el mobiliario de forma adecuada (sillas, mesas y postura al trabajar frente al equipo).
- Evitar bloqueos de salidas de emergencia o pasillos.

Reglamento de uniforme

No aplica

Uso adecuado del equipo y materiales

- Utilizar el equipo de cómputo conforme a las indicaciones del docente o personal responsable.
- No modificar configuraciones predeterminadas, como el fondo de escritorio o la instalación de programas sin autorización.
- Mantener limpios y en buen estado los equipos y mobiliario del aula.
- Evitar el consumo de alimentos y bebidas cerca del equipo de cómputo.
- Reportar inmediatamente cualquier falla, daño o mal funcionamiento al responsable del laboratorio o docente.
- Guardar correctamente los archivos generados durante la sesión en las ubicaciones designadas (USB, nube, etc.).
- Cerrar adecuadamente todos los programas antes de apagar el equipo.
- Apagar correctamente la computadora al finalizar la sesión, siguiendo el procedimiento establecido.
- No desconectar ni manipular cables, dispositivos periféricos o componentes internos del equipo sin autorización.
- Utilizar de forma responsable y exclusiva los recursos con fines académicos.

Manejo y disposición de residuos peligrosos

No aplica

Procedimientos en caso de emergencia

En caso de presentarse una incidencia seguir las indicaciones del personal a cargo.

RELACIÓN DE PRÁCTICAS DE LABORATORIO POR ELEMENTO DE COMPETENCIA

Elemento de Competencia al que pertenece la práctica	<p>I</p> <p>Elaborar programas sencillos, en un entorno de desarrollo integrado, con el fin de sentar las bases para la solución de problemas más complejos, aplicando los principios de una buena programación y poniendo en práctica las habilidades de análisis de problemas, el trabajo en equipo y la responsabilidad.</p>
---	--

PRÁCTICA	NOMBRE	COMPETENCIA
Práctica No. 1	Programas en Java para entrada y salida de datos por consola.	Desarrollar programas en lenguaje Java que utilicen correctamente las instrucciones de entrada y salida por consola, aplicando estructuras básicas

		de programación, como lectura de datos, operaciones aritméticas y presentación de resultados, fomentando el trabajo colaborativo y la comunicación efectiva para la solución de problemas en equipo.
--	--	--

Elemento de Competencia al que pertenece la práctica	II
	Construir programas aplicando estructuras de control selectivas y repetitivas, en un entorno de desarrollo integrado con el fin de ofrecer soluciones eficientes a problemas más complejos, aplicando los principios de una buena programación y poniendo en práctica las habilidades de análisis de problemas, el trabajo en equipo y la responsabilidad.

PRÁCTICA	NOMBRE	COMPETENCIA
Práctica No. 2	Programas en Java aplicando la estructura selectiva if...else.	Aplicar la estructura selectiva if...else en el desarrollo de programas en Java para resolver problemas que requieren tomar decisiones en función de condiciones establecidas por el usuario o el sistema, asumiendo con responsabilidad y autonomía el desarrollo de soluciones funcionales y confiables.
Práctica No. 3	Práctica de programación en Java aplicando la estructura selectiva switch.	Implementar estructuras de control selectivas tipo switch en lenguaje Java para resolver problemas donde se deben realizar acciones distintas según el valor de una variable, desarrollando programas organizados y funcionales.
Práctica No. 4	Programas en Java aplicando la estructura repetitiva for.	Implementar la estructura repetitiva for en programas desarrollados en lenguaje Java, resolviendo problemas mediante ciclos controlados por contador, fomentando la lógica de repetición en tareas automatizadas, fortaleciendo el trabajo colaborativo para compartir ideas, depurar código y construir soluciones de manera conjunta.
Práctica No. 5	Ejercicios en Java aplicando la estructura repetitiva while	Implementar la estructura repetitiva while en programas desarrollados en lenguaje Java, para resolver problemas en los que

		no se conoce previamente el número de iteraciones, permitiendo desarrollar lógica algorítmica dinámica.
Práctica No. 6	Práctica de programación en Java aplicando la estructura repetitiva do while	Implementar la estructura repetitiva do while en programas desarrollados en lenguaje Java, resolviendo problemas que requieren al menos una ejecución del bloque de instrucciones, fomentando la lógica de repetición controlada por condición final.

Elemento de Competencia al que pertenece la práctica	III
	Desarrollar programas en un entorno de desarrollo integrado, usando arreglos, como una solución a problemas de representación de datos que surgen en las organizaciones y diseñar propuestas donde se apliquen los métodos de clase, los principios de una buena programación y las habilidades de análisis de problemas, el trabajo en equipo y la responsabilidad.

PRÁCTICA	NOMBRE	COMPETENCIA
Práctica No. 7	Ejercicios en Java usando arreglos unidimensionales	Implementar el uso de arreglos unidimensionales en programas desarrollados en lenguaje Java para almacenar, procesar y manipular conjuntos de datos, fomentando la comprensión del manejo eficiente de estructuras de datos, así como la capacidad de analizar problemas, estructurar soluciones lógicas y tomar decisiones durante el desarrollo de software.
Práctica No. 8	Programas en Java usando arreglos bidimensionales	Implementar el uso de arreglos bidimensionales en programas desarrollados en lenguaje Java para almacenar, procesar y validar conjuntos de datos estructurados, fortaleciendo la comprensión del manejo eficiente de estructuras de datos.
Práctica No. 9	Programas en Java usando métodos de clase o	Implementar el uso de arreglos bidimensionales en programas



	estáticos.	desarrollados en lenguaje Java para almacenar, procesar y validar conjuntos de datos estructurados, fortaleciendo la comprensión del manejo eficiente de estructuras de datos.
--	------------	--



PRÁCTICAS



NOMBRE DE LA PRÁCTICA	Programas en Java para entrada y salida de datos por consola.
COMPETENCIA DE LA PRÁCTICA	Desarrollar programas en lenguaje Java que utilicen correctamente las instrucciones de entrada y salida por consola, aplicando estructuras básicas de programación, como lectura de datos, operaciones aritméticas y presentación de resultados.

FUNDAMENTO TEÓRICO

El uso de entrada y salida estándar en Java es fundamental para la interacción entre el usuario y el programa. A través de la clase Scanner, se permite leer datos desde el teclado y con System.out.println se presenta información en pantalla. Estas funcionalidades son esenciales para construir aplicaciones que respondan dinámicamente a las acciones del usuario.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora con entorno de desarrollo Java (NetBeans, IntelliJ IDEA o similar)
- Java Development Kit (JDK) instalado
- Acceso a plataforma educativa para entrega de evidencia
- Documento de entrega en formato PDF con ejercicios resueltos

PROCEDIMIENTO O METODOLOGÍA

Desarrollar los siguientes programas en JAVA usando la herramienta NetBeans u otra equivalente.

Cada ejercicio deberá ser implementado en una clase independiente, manteniéndose dentro del mismo proyecto.. El reporte debe incluir portada, descripción del ejercicio, imagen del código completo y de su ejecución. El documento se entregará en formato PDF a través de la plataforma correspondiente

- 1.- Escriba un programa que imprima en la pantalla su nombre completo en una línea y en la línea siguiente su fecha de nacimiento.
- 2.- Desarrolle un proyecto que le solicite su nombre e imprima en pantalla: "Bienvenido y el nombre ingresado."
- 3.- Realizar un programa que permita obtener el doble de un número leído a través del teclado.
- 4.- Realizar un programa que permita obtener el cuadrado de un número leído a través del teclado.
- 5.- Se quiere analizar la capacidad de cálculo de una computadora, por lo que se quiere un programa que solucione el siguiente problema:
Leer 5 números enteros, y calcular lo siguiente:
 - a) La suma de los números.
 - b) El Promedio de los 5 números.
 - c) El producto de los 5 números
- 6.- Un vendedor recibe un sueldo base más un 10% extra por comisión de sus ventas, el vendedor

desea saber cuánto dinero obtendrá por concepto de comisiones por las tres ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo base y comisiones.

7.- Un maestro desea saber que porcentaje de hombres y que porcentaje de mujeres hay en un grupo de estudiantes.

8.- Elabore un programa en JAVA que le permita convertir de pesos a dólares, debe de leer el valor actual del dólar.

9.- Elabore un programa en JAVA para calcular el nuevo salario de un obrero si obtuvo un incremento del 25% sobre su salario anterior.

10.- En un supermercado se vende el kg de manzanas a \$30 pesos, el de plátano a \$15 y el de uva a \$50.

Desarrolle un programa que calcule y presente por pantalla el subtotal a pagar y el total a pagar (con IVA del 16%) cuando una persona compre x cantidad de kilos de cada fruta mencionada.

El programa debe solicitar el kilo de cada fruta que desea el cliente y realizar los cálculos necesarios para obtener el subtotal y total que debe pagar.

RESULTADOS ESPERADOS

- El estudiante es capaz de desarrollar programas con entrada y salida estándar en Java.
- El estudiante puede aplicar operaciones básicas y presentar resultados correctamente.
- Entrega de un reporte completo con la solución de cada ejercicio.

ANÁLISIS DE RESULTADOS

1. ¿Qué instrucción permite leer datos desde el teclado en Java?
2. ¿Cuál fue la dificultad principal al desarrollar los programas?
3. ¿Qué aprendiste sobre el manejo de datos en Java?
4. ¿En qué escenarios reales se pueden aplicar estos programas?

CONCLUSIONES Y REFLEXIONES

El dominio de las instrucciones básicas de entrada y salida en Java es esencial para la programación orientada a objetos. Estos ejercicios permiten al estudiante afianzar conocimientos fundamentales y visualizar aplicaciones prácticas en su entorno profesional.

ACTIVIDADES COMPLEMENTARIAS

Investigar sobre el tema para fortalecer el aprendizaje.

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación - Desarrollo correcto de los ejercicios.

	- Cumplimiento del formato de entrega. - Claridad en la presentación del reporte
Rúbricas o listas de cotejo para valorar desempeño	Rubrica para la solución de ejercicios.
Formatos de reporte de prácticas	Archivo PDF

NOMBRE DE LA PRÁCTICA	Programas en Java aplicando la estructura selectiva if...else.
COMPETENCIA DE LA PRÁCTICA	Aplicar la estructura selectiva if...else en el desarrollo de programas en Java para resolver problemas que requieren tomar decisiones en función de condiciones establecidas por el usuario o el sistema.

FUNDAMENTO TEÓRICO
La estructura selectiva if...else permite que un programa tome decisiones durante su ejecución. Dependiendo del resultado de una condición lógica, se ejecuta uno u otro bloque de instrucciones. Es una herramienta fundamental en la programación para el control del flujo de ejecución y la solución de problemas que implican condiciones.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
- Computadora con entorno de desarrollo Java (NetBeans, IntelliJ IDEA o similar) - Java Development Kit (JDK) instalado - Plataforma educativa para entrega de evidencias - Documento PDF con la descripción, código fuente y evidencia de ejecución de cada ejercicio

PROCEDIMIENTO O METODOLOGÍA
Desarrollar los siguientes programas en JAVA usando la herramienta NetBeans u otra equivalente.
Cada ejercicio deberá ser implementado en una clase independiente, manteniéndose dentro del mismo proyecto.. El reporte debe incluir portada, descripción del ejercicio, imagen del código completo y de su ejecución. El documento se entregará en formato PDF a través de la plataforma correspondiente
Ejercicios:
1. Leer edad y mostrar el mensaje "Mayor de edad" si edad es igual o mayor de 18 años.
2. Leer edad y mostrar el mensaje "Mayor de edad" si edad es igual o mayor de 18 años, y "Menor de edad" si es menor.
3. Leer una calificación y mostrar mensaje 'Aprobado' si calificación ≥ 70 .
4. Mostrar el mensaje "Aprobado" si el promedio de 3 calificaciones es mayor a 6.

5. Determinar si un número es positivo, negativo o neutro.
6. Calcular descuento del 15% si la venta es > 500 pesos, o 10% si es ≤ 500.
7. Leer 2 números e imprimir el mayor.
8. Leer 4 números e imprimir el mayor.
9. Leer 3 números e imprimir el mayor y el menor.

RESULTADOS ESPERADOS

- Aplicación correcta de la estructura if...else en distintos contextos.
- Desarrollo de lógica condicional en problemas cotidianos.
- Elaboración de un reporte funcional y ordenado con los ejercicios solicitados.

ANÁLISIS DE RESULTADOS

1. ¿En qué ejercicios fue necesario usar una estructura condicional anidada?
2. ¿Cómo afecta el orden de evaluación de condiciones al resultado del programa?
3. ¿Qué tipos de errores lógicos encontraste al desarrollar los programas?
4. ¿Qué utilidad tienen las estructuras condicionales en aplicaciones reales?

CONCLUSIONES Y REFLEXIONES

El uso de estructuras condicionales como if...else es esencial en la programación, ya que permite adaptar el comportamiento del software en función de diferentes situaciones. Su correcta implementación mejora la toma de decisiones automatizadas, lo cual es común en múltiples entornos laborales y profesionales.

ACTIVIDADES COMPLEMENTARIAS

1. Crear un programa que determine si un año es bisiesto.

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	-Desarrollo correcto de los ejercicios. -Cumplimiento del formato de entrega. - Claridad en la presentación del reporte
Rúbricas o listas de cotejo para valorar desempeño	Rubrica para la solución de ejercicios.
Formatos de reporte de prácticas	Archivo PDF

NOMBRE DE LA PRÁCTICA	Práctica de programación en Java aplicando la estructura selectiva switch.
COMPETENCIA DE LA PRÁCTICA	Implementar estructuras de control selectivas tipo switch en lenguaje Java para resolver problemas donde se deben realizar acciones distintas según el valor de una variable, desarrollando programas organizados y funcionales.

FUNDAMENTO TEÓRICO
La estructura selectiva switch en Java permite ejecutar diferentes bloques de código dependiendo del valor de una expresión. Es útil cuando se deben evaluar múltiples posibles valores de una sola variable. Su uso mejora la claridad del código y evita múltiples estructuras if...else anidadas. Es especialmente efectiva para casos con opciones limitadas y predefinidas

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> - Computadora con entorno de desarrollo Java (NetBeans, IntelliJ IDEA u otro) - Java Development Kit (JDK) - Plataforma educativa para entregar el reporte - Documento en formato PDF con ejercicios resueltos, código e imágenes de ejecución

PROCEDIMIENTO O METODOLOGÍA
<p>Desarrollar los siguientes programas en JAVA usando la herramienta NetBeans u otra equivalente. Cada ejercicio deberá ser implementado en una clase independiente, manteniéndose dentro del mismo proyecto.. El reporte debe incluir portada, descripción del ejercicio, imagen del código completo y de su ejecución. El documento se entregará en formato PDF a través de la plataforma correspondiente</p> <p>Ejercicios:</p> <ol style="list-style-type: none"> 1. Programa que muestre un mensaje según la calificación (100: Excelente, 90: Muy bien, 80: Bien, 70: Regular, 60/50: Muy mal, otro: No válido). 2. Programa que evalúe una variable con el número de mes (1 al 12) y muestre su nombre. Si es inválido, mostrar advertencia. 3. Programa que indique la cantidad de días según el número del mes. 4. Calculadora básica que solicite dos números y un operador (+, -, *, /), realice la operación y muestre el resultado.

RESULTADOS ESPERADOS
<ul style="list-style-type: none"> - Aplicación correcta de la estructura switch para tomar decisiones múltiples. - Identificación de los casos válidos y uso de la cláusula default. - Presentación clara y estructurada de las soluciones y salidas esperadas.

ANÁLISIS DE RESULTADOS

1. ¿Qué ventajas ofrece el uso de switch sobre if...else en estos ejercicios?
2. ¿Qué ocurre si no se incluye la instrucción break en cada caso?
3. ¿Cómo manejaste valores inválidos o no contemplados?
4. ¿Cuál fue el ejercicio más complejo y por qué?

CONCLUSIONES Y REFLEXIONES

La estructura switch permite seleccionar eficientemente entre múltiples opciones de ejecución según el valor de una variable. Este enfoque es especialmente útil cuando se trabaja con datos categóricos o numéricos limitados. Su implementación adecuada es fundamental para construir aplicaciones lógicas, limpias y mantenibles.

ACTIVIDADES COMPLEMENTARIAS

Investigar sobre el tema para fortalecer el aprendizaje.

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	<ul style="list-style-type: none"> - Desarrollo correcto de los ejercicios. - Cumplimiento del formato de entrega. - Claridad en la presentación del reporte
Rúbricas o listas de cotejo para valorar desempeño	Rubrica para la solución de ejercicios.
Formatos de reporte de prácticas	Archivo PDF

NOMBRE DE LA PRÁCTICA

Programas en Java aplicando la estructura repetitiva for.

COMPETENCIA DE LA PRÁCTICA

Implementar la estructura repetitiva for en programas desarrollados en lenguaje Java, resolviendo problemas mediante ciclos controlados por contador, fomentando la lógica de repetición en tareas automatizadas, fortaleciendo el trabajo colaborativo para compartir ideas, depurar código y construir soluciones de manera conjunta

FUNDAMENTO TEÓRICO

La estructura repetitiva for permite ejecutar un bloque de instrucciones un número determinado de veces. Está compuesta por tres partes principales: inicialización, condición y actualización. Es ampliamente utilizada cuando se conoce de antemano cuántas veces debe repetirse una

operación, lo que la hace ideal para recorrer rangos numéricos, listas o realizar tareas repetitivas de forma controlada.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

- Computadora personal o laptop
- Entorno de desarrollo Java (NetBeans, IntelliJ IDEA, Eclipse, etc.)
- Java Development Kit (JDK) actualizado
- Plataforma de entrega institucional
- Archivo PDF con reporte de ejercicios desarrollados, código fuente y evidencia de ejecución

PROCEDIMIENTO O METODOLOGÍA

Desarrollar los siguientes programas en JAVA usando la herramienta NetBeans u otra equivalente.

Cada ejercicio deberá ser implementado en una clase independiente, manteniéndose dentro del mismo proyecto.. El reporte debe incluir portada, descripción del ejercicio, imagen del código completo y de su ejecución. El documento se entregará en formato PDF a través de la plataforma correspondiente.

Ejercicios:

1. Imprimir 10 veces el mensaje "Hola".
2. Imprimir 30 veces su nombre.
3. Imprimir los números del 1 al 20.
4. Imprimir los números del 1 al 1000.
5. Imprimir los números pares entre 1 y 100.
6. Imprimir los números impares entre 1 y 100.
7. Leer 10 números ingresados por el usuario.
8. Leer e imprimir la suma de 10 números.
9. Leer 10 números y calcular su promedio.
10. Capturar los siguientes datos de 5 clientes: Nombre, Dirección y Teléfono.
11. Imprimir las tablas de multiplicar del 1 al 10
12. Solicita al usuario un número entero positivo e imprime su tabla de multiplicar del 1 al 10.
13. Solicita un número entero positivo y determina si es un número primo.
14. Solicita un número entero positivo y muestra su factorial.

RESULTADOS ESPERADOS

- El estudiante implementa correctamente ciclos for para controlar repeticiones.
- Se muestra dominio del flujo de control iterativo en Java.
- Se obtiene un reporte estructurado con cada ejercicio documentado y ejecutado correctamente.

ANÁLISIS DE RESULTADOS

1. ¿Qué beneficios ofrece el uso del ciclo for frente a repetir instrucciones manualmente?
2. ¿Cuál fue el ejercicio más desafiante y por qué?
3. ¿Qué errores comunes se presentan al manipular contadores en ciclos?
4. ¿En qué situaciones reales sería útil automatizar tareas usando for?

CONCLUSIONES Y REFLEXIONES

El ciclo for es una herramienta fundamental en la programación para ejecutar tareas repetitivas de manera controlada. Su aplicación permite automatizar procesos, mejorar la eficiencia del código y facilitar el desarrollo de soluciones a problemas prácticos. Comprender y aplicar esta estructura es clave en la formación de programadores profesionales.

ACTIVIDADES COMPLEMENTARIAS

Desarrollar un programa en JAVA, que permita trabajar con **10 series** de números, deberá cumplir con los siguientes requisitos:

- 1.- Las series de números usted los debe de inventar (Ejemplo: serie de números naturales, serie de números primos, serie de números impares etc.
- 2.- Presentar al usuario un Menú con las diferentes opciones de series, para que seleccione cual desea ver.
- 3.- El usuario debe de indicar cuantos números de las series desea.
4. Debe utilizar ciclo Do-While
- 5.- Debe de utilizar Switch
- 6- Probar todas las series funcionen correctamente
- 7.- Presentar la serie de números en horizontal para aprovechar mejor el espacio
- 8.- Cuidar la buena presentación de la información que presentara en pantalla.
9. El programa debe de estar funcionando mientras el usuario no lo indique. Conveniente tener la opción 11 para salir

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	-Desarrollo correcto de los ejercicios. -Cumplimiento del formato de entrega. - Claridad en la presentación del reporte
Rúbricas o listas de cotejo para valorar desempeño	Rubrica para la solución de ejercicios.
Formatos de reporte de prácticas	Archivo PDF

NOMBRE DE LA PRÁCTICA

Ejercicios en Java aplicando la estructura repetitiva

COMPETENCIA DE LA PRÁCTICA	while
	Implementar la estructura repetitiva while en programas desarrollados en lenguaje Java, para resolver problemas en los que no se conoce previamente el número de iteraciones, permitiendo desarrollar lógica algorítmica dinámica.

FUNDAMENTO TEÓRICO

La estructura repetitiva while en Java permite ejecutar un bloque de instrucciones mientras se cumpla una condición lógica. A diferencia del ciclo for, el número de repeticiones no está definido previamente, lo que lo hace ideal para situaciones donde el fin del proceso depende de una entrada del usuario o de una condición variable. Es comúnmente utilizada en validaciones, menús interactivos y acumuladores de datos.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

Computadora o laptop con entorno de desarrollo Java (NetBeans, IntelliJ, Eclipse, etc.)
Java Development Kit (JDK) instalado y configurado
Plataforma institucional para entrega de reportes
Documento de reporte en formato PDF con: portada, ejercicios desarrollados, código fuente e imágenes de ejecución

PROCEDIMIENTO O METODOLOGÍA

Desarrollar los siguientes programas en JAVA, usando la estructura repetitive While, en la herramienta NetBeans o cualquier otra de su elección.

Cada ejercicio deberá ser implementado en una clase independiente, manteniéndose dentro del mismo proyecto.

El reporte debe incluir portada, descripción del ejercicio, imagen del código completo y de su ejecución. El documento se entregará en formato PDF a través de la plataforma correspondiente.. Generar reporte de entrega que incluya: Portada, descripción del ejercicio, imagen del código completo y de la ejecución correcta. Generar documento PDF y subirlo a la plataforma.

Ejercicios:

1. Imprimir 10 veces el mensaje "HOLA".
2. Imprimir los números positivos del 1 al 20.
3. Leer 10 números.
4. Escribir la frase "¿Deseas continuar S/N?" mientras la respuesta sea 'S'.
5. Capturar ventas del día sin conocer cuántas habrá. Preguntar al usuario si desea capturar otra. Al final, mostrar la suma total de ventas.
6. Capturar Nombre, Dirección y Teléfono de clientes sin saber cuántos serán. Al final, indicar cuántos clientes fueron ingresados.
7. Solicitar un número entero positivo y sumar todos los números pares desde 1 hasta dicho número.
8. Generar una serie de números primos, donde la cantidad de números es definida por el

usuario.

RESULTADOS ESPERADOS

- Uso correcto de la estructura while en situaciones que requieren repetición condicional.
- Ejecución funcional de los programas con lógica controlada por el usuario.
- Desarrollo de habilidades para resolver problemas mediante programación iterativa sin número fijo de ciclos

ANÁLISIS DE RESULTADOS

- ¿En qué ejercicios fue más útil la estructura while frente al ciclo for?
- ¿Qué condiciones fueron más difíciles de controlar?
- ¿Cómo verificaste que el ciclo se detuviera adecuadamente?
- ¿Qué aprendiste sobre el manejo dinámico del flujo de ejecución?

CONCLUSIONES Y REFLEXIONES

El uso del ciclo while permite crear programas que se adaptan dinámicamente a las decisiones del usuario o a condiciones que se evalúan durante la ejecución. Esta flexibilidad lo convierte en una herramienta esencial para la creación de menús, acumuladores, y estructuras de control no lineales. Su dominio es clave para resolver problemas del mundo real donde el número de iteraciones no se puede predecir de antemano

ACTIVIDADES COMPLEMENTARIAS

1. Crear un menú interactivo que permita realizar operaciones matemáticas básicas hasta que el usuario decida salir.
2. Escribir un programa que valide si una contraseña ingresada es correcta, permitiendo múltiples intentos hasta acertar.

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	-Desarrollo correcto de todos los ejercicios y prueba de cada uno de ellos. -Cumplimiento del formato de entrega. - Claridad en la presentación del reporte
Rúbricas o listas de cotejo para valorar desempeño	Rubrica para la solución de ejercicios.
Formatos de reporte de	Archivo PDF

prácticas

NOMBRE DE LA PRÁCTICA

Práctica de programación en Java aplicando la estructura repetitiva do while

COMPETENCIA DE LA PRÁCTICA

Implementar la estructura repetitiva do while en programas desarrollados en lenguaje Java, resolviendo problemas que requieren al menos una ejecución del bloque de instrucciones, fomentando la lógica de repetición controlada por condición final.

FUNDAMENTO TÉCNICO

La estructura do while en Java permite ejecutar un bloque de código al menos una vez, ya que la condición se evalúa al final del ciclo. Es útil en situaciones donde se requiere que el cuerpo del ciclo se ejecute al menos una vez, como en menús interactivos, validaciones de entrada o cuando el número de iteraciones no es conocido de antemano.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS

Computadora o laptop con entorno de desarrollo Java (NetBeans, IntelliJ, Eclipse, etc.)
Java Development Kit (JDK) actualizado
Plataforma educativa institucional
Reporte en formato PDF con portada, código, descripción y evidencia de ejecución de cada ejercicio

PROCEDIMIENTO O METODOLOGÍA

Desarrollar los siguientes programas en JAVA, usando la estructura repetitiva do while en la herramienta NetBeans o cualquier otra de su elección. Cada ejercicio debe implementarse en una clase independiente dentro del mismo proyecto.

El reporte debe incluir portada, descripción del ejercicio, imagen del código completo y de su ejecución. El documento se entregará en formato PDF a través de la plataforma correspondiente.

Ejercicios:

1. Solicitar al usuario un número entero positivo y sumar todos los números pares desde 1 hasta ese número.
2. Leer 10 números y sumar solo los pares ingresados. Mostrar la suma.
3. Imprimir la serie Fibonacci hasta la cantidad indicada por el usuario.
4. Imprimir los factoriales desde 1 hasta n.
5. Generar una serie de números primos según la cantidad indicada por el usuario.

RESULTADOS ESPERADOS

- Aplicación correcta de la estructura do while.
Solución de cada uno de los ejercicios indicados.
- Prueba del funcionamiento correcto de cada uno de los ejercicios.

ANÁLISIS DE RESULTADOS

1. ¿Qué diferencia fundamental hay entre un ciclo do while y un ciclo while?
2. ¿En qué ejercicios fue indispensable usar al menos una ejecución antes de evaluar la condición?
3. ¿Cómo aseguraste que los programas no entraran en bucles infinitos?
4. ¿Qué errores lógicos encontraste durante el desarrollo y cómo los solucionaste?

CONCLUSIONES Y REFLEXIONES

El ciclo do while es ideal para situaciones donde se necesita ejecutar una acción al menos una vez antes de validar una condición. Su dominio permite crear programas interactivos y robustos, especialmente útiles para validaciones y procesamiento de datos de entrada en aplicaciones reales. Su correcta implementación fortalece las competencias en estructuras de control repetitivas.

ACTIVIDADES COMPLEMENTARIAS

1. Crear un programa que muestre un menú hasta que el usuario seleccione la opción 'Salir'.
2. Implementar un sistema de captura de temperaturas que permita calcular el promedio al final, mientras el usuario desee continuar.

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	-Desarrollo correcto de los ejercicios. -Cumplimiento del formato de entrega. - Claridad en la presentación del reporte
Rúbricas o listas de cotejo para valorar desempeño	Rubrica para la solución de ejercicios.
Formatos de reporte de prácticas	Archivo PDF

NOMBRE DE LA PRÁCTICA	Ejercicios en Java usando arreglos unidimensionales
COMPETENCIA DE LA PRÁCTICA	Implementar el uso de arreglos unidimensionales en programas desarrollados en lenguaje Java para almacenar, procesar y manipular conjuntos de datos, fomentando la comprensión del manejo eficiente de estructuras de datos, así como la capacidad de analizar problemas, estructurar soluciones lógicas y tomar decisiones durante el desarrollo de software.

FUNDAMENTO TEÓRICO
Los arreglos unidimensionales son estructuras de datos que permiten almacenar múltiples valores del mismo tipo bajo un solo identificador. En Java, los arreglos se utilizan para organizar datos secuenciales como listas de números o registros. El uso de arreglos permite recorrer, modificar y analizar datos de forma eficiente mediante ciclos.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> • Computadora con entorno de desarrollo Java (NetBeans, IntelliJ, Eclipse, etc.) • Java Development Kit (JDK) actualizado • Acceso a plataforma educativa para entrega • Reporte en formato PDF con código fuente, descripciones e imágenes de ejecución

PROCEDIMIENTO O METODOLOGÍA
<p>Desarrollar los siguientes programas en JAVA, usando arreglos unidimensionales en la herramienta NetBeans o equivalente.</p> <p>Cada ejercicio deberá implementarse en una clase independiente dentro del mismo proyecto.</p> <p>El reporte debe incluir portada, descripción del ejercicio, imagen del código completo y de su ejecución. El documento se entregará en formato PDF a través de la plataforma institucional.</p> <p>Ejercicios:</p> <ol style="list-style-type: none"> 1. Leer 10 números, almacenarlos en un arreglo e imprimirlos. 2. Almacenar en un arreglo 15 números pares e imprimir su contenido. 3. Leer 8 números, almacenarlos y mostrar solo los pares del arreglo. 4. Leer 10 números, verificar si un número solicitado al usuario fue ingresado. 5. Leer 15 números, encontrar el menor y su posición. 6. Generar 1000 números aleatorios entre 0-100, almacenarlos e imprimirlos. 7. Leer 8 números, contar cuántos son positivos, negativos y neutros, y mostrar el arreglo completo. 8. Leer 10 números, mostrar listados separados de positivos, negativos y neutros, junto con el arreglo completo.

RESULTADOS ESPERADOS

Parámetros para evaluar o datos a recolectar

ANÁLISIS DE RESULTADOS

- ¿Cómo ayuda el uso de arreglos a simplificar la manipulación de datos?
- ¿Qué dificultades encontraste al trabajar con índices y validaciones?
- ¿Cuál fue el ejercicio más retador y cómo lo solucionaste?
- ¿Qué ventajas observas en almacenar datos en arreglos respecto a variables individuales?

CONCLUSIONES Y REFLEXIONES

El manejo de arreglos unidimensionales permite trabajar de forma más organizada y eficiente con grandes volúmenes de datos. Su uso es esencial para el desarrollo de programas que requieren almacenar, buscar o clasificar múltiples elementos. Estas habilidades son fundamentales en áreas como análisis de datos, simulaciones y desarrollo de software comercial.

ACTIVIDADES COMPLEMENTARIAS

1. Crear un programa que permita ordenar los elementos de un arreglo de menor a mayor.
2. Desarrollar una función que calcule la media, mediana y moda de los elementos almacenados en un arreglo.

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	-Desarrollo correcto de todos los ejercicios y prueba de cada uno de ellos. -Cumplimiento del formato de entrega. - Claridad en la presentación del reporte
Rúbricas o listas de cotejo para valorar desempeño	Rubrica para la solución de ejercicios.
Formatos de reporte de prácticas	Archivo PDF

NOMBRE DE LA PRÁCTICA	Programas en Java usando arreglos bidimensionales
COMPETENCIA DE LA PRÁCTICA	Implementar el uso de arreglos bidimensionales en programas desarrollados en lenguaje Java para almacenar, procesar y validar conjuntos de datos estructurados, fortaleciendo la comprensión del manejo eficiente de estructuras de datos.

FUNDAMENTO TEÓRICO
Los arreglos bidimensionales permiten representar estructuras matriciales, es decir, datos organizados en filas y columnas. En Java, se utilizan para almacenar múltiples valores en una estructura de tabla que puede ser accedida y manipulada por medio de índices. Este tipo de estructura es útil en aplicaciones que requieren modelar datos tabulares, realizar cálculos o validaciones sobre conjuntos de valores relacionados.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> - Computadora con entorno de desarrollo Java (NetBeans, IntelliJ IDEA, Eclipse, etc.) - Java Development Kit (JDK) actualizado - Plataforma institucional para entrega de evidencias - Reporte en formato PDF con portada, descripción de ejercicios, código fuente y evidencia de ejecución

PROCEDIMIENTO O METODOLOGÍA
<p>Desarrollar los siguientes programas en JAVA usando arreglos bidimensionales en la herramienta NetBeans o cualquier otra de su elección.</p> <p>Cada ejercicio debe implementarse en una clase independiente dentro del mismo proyecto.</p> <p>El reporte debe incluir portada, descripción del ejercicio, imagen del código completo y de su ejecución. El documento se entregará en formato PDF a través de la plataforma correspondiente.</p> <p>Ejercicios:</p> <ol style="list-style-type: none"> 1. Llenar una matriz de 3x3 con números leídos a través del teclado. Imprimir la matriz. 2. Llenar una matriz de 4x3 con el número 2. Imprimir la matriz. 3. Llenar una matriz de 4x3 con números pares. Imprimir la matriz. 4. Llenar una matriz de 2x4 con números impares introducidos por el teclado. Validar que el valor sea impar.

RESULTADOS ESPERADOS
<ul style="list-style-type: none"> - Implementación funcional de arreglos bidimensionales para el almacenamiento y validación de datos. - Aplicación de lógica condicional para validar si los datos ingresados cumplen con criterios

específicos (pares, impares, primos).
- Entrega de reporte con evidencia completa de ejecución de los ejercicios.

ANÁLISIS DE RESULTADOS

1. ¿Cuál fue la principal dificultad al validar condiciones dentro de la matriz?
2. ¿Qué lógica utilizaste para verificar si un número es primo o impar?
3. ¿Cómo estructuraste el uso de índices para recorrer filas y columnas eficientemente?
4. ¿Qué beneficios ofrece el uso de arreglos bidimensionales frente a múltiples arreglos unidimensionales?

CONCLUSIONES Y REFLEXIONES

El uso de arreglos bidimensionales es fundamental para modelar estructuras de datos complejas como tablas o matrices numéricas. Permite organizar datos relacionados en múltiples dimensiones y aplicar validaciones que responden a requerimientos específicos. Este conocimiento es aplicable en múltiples contextos como bases de datos, gráficos, simulaciones y análisis de información.

ACTIVIDADES COMPLEMENTARIAS

1. Crear un programa que sume los valores de cada fila de una matriz 4x4.
2. Desarrollar un programa que almacene valores en una matriz 5x5 y luego intercambie la primera y última fila.

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	-Desarrollo correcto de los ejercicios. -Cumplimiento del formato de entrega. - Claridad en la presentación del reporte
Rúbricas o listas de cotejo para valorar desempeño	Rubrica para la solución de ejercicios.
Formatos de reporte de prácticas	Archivo PDF

NOMBRE DE LA PRÁCTICA	Programas en Java usando métodos de clase o estáticos.
COMPETENCIA DE LA PRÁCTICA	Implementar métodos de clase (estáticos) en programas desarrollados en Java para modularizar soluciones, reutilizar código y resolver problemas prácticos mediante el diseño de funciones específicas, fortaleciendo también la capacidad de análisis lógico y toma de decisiones en la resolución de problemas.

FUNDAMENTO TEÓRICO
En Java, los métodos estáticos (static) pertenecen a la clase en lugar de a una instancia de la clase. Esto significa que pueden ser llamados sin crear un objeto. Son útiles para operaciones que no dependen de los valores de instancia, como cálculos generales o utilidades. El uso de métodos estáticos mejora la organización del código, promueve la reutilización de funciones y facilita la implementación de soluciones modulares.

MATERIALES, EQUIPAMIENTO Y/O REACTIVOS
<ul style="list-style-type: none"> - Computadora con entorno de desarrollo Java (NetBeans, IntelliJ IDEA, Eclipse, etc.) - Java Development Kit (JDK) instalado y actualizado - Plataforma institucional de entrega - Documento de entrega en formato PDF con código fuente, explicación, imágenes y conclusiones

PROCEDIMIENTO O METODOLOGÍA
<p>Desarrollar el siguiente programa en JAVA, utilizando la herramienta NetBeans u otra de su elección. El proyecto debe cumplir con las siguientes indicaciones:</p> <ul style="list-style-type: none"> - Investigar qué son los métodos static en Java. - Explicar cuándo deben usarse. - Incluir un ejemplo básico de un método static. - Crear un proyecto que contenga al menos dos métodos static: uno para calcular el área y otro para calcular el perímetro de un rectángulo. - El usuario debe ingresar la base y altura, y seleccionar si desea ver el área o el perímetro. - El programa debe funcionar de manera cíclica. - Redactar una conclusión sobre el tema. - Incluir evidencia de trabajo (capturas de pantalla) en el reporte. <p>El reporte debe incluir: portada, descripción del ejercicio, imagen del código completo y de su ejecución. El documento se entregará en formato PDF mediante la plataforma institucional.</p>

RESULTADOS ESPERADOS
- Comprensión del concepto y uso de métodos estáticos en Java.

- Implementación correcta de funciones reutilizables para resolver cálculos específicos.
- Generación de un programa funcional y documentado.
- Capacidad de integrar entradas del usuario y lógica condicional en ciclos controlados.

ANÁLISIS DE RESULTADOS

1. ¿Cuáles son las ventajas de utilizar métodos estáticos en un programa?
2. ¿Qué dificultades se presentaron al implementar funciones con parámetros?
3. ¿Cómo se puede mejorar la estructura del programa para hacerlo más legible y reutilizable?
4. ¿Qué impacto tiene el uso de métodos en la organización general del proyecto?

CONCLUSIONES Y REFLEXIONES

El uso de métodos estáticos permite encapsular operaciones que no requieren estado de objeto, promoviendo la modularidad y claridad del código. Su aplicación en problemas como el cálculo de área o perímetro permite al estudiante desarrollar lógica estructurada, controlar flujos mediante ciclos y mejorar sus habilidades en diseño de funciones y reutilización de código.

ACTIVIDADES COMPLEMENTARIAS

1. Crear un método estático que reciba una cadena de texto y devuelva la cantidad de vocales que contiene.
2. Implementar un menú interactivo que utilice métodos estáticos para realizar operaciones básicas (suma, resta, multiplicación, división).

EVALUACIÓN Y EVIDENCIAS DE APRENDIZAJE

Criterios de evaluación	-Desarrollo correcto de los ejercicios. -Cumplimiento del formato de entrega. - Claridad en la presentación del reporte
Rúbricas o listas de cotejo para valorar desempeño	Rubrica para la solución de ejercicios.
Formatos de reporte de prácticas	Archivo PDF

FUENTES DE INFORMACIÓN

- Byspel. (2018, julio 3). *Arreglos de tipo string en Java - Manejo de cadenas de arreglos en Java* [Video]. YouTube. <https://youtu.be/JsF5O6QeefM>
- Ceballos Sierra, F. J. (2015). *Java 2: lenguaje y aplicaciones*. RA-MA Editorial. <https://elibro.net/es/lc/ues/titulos/62458>
- Códigos de Programación. (2017, marzo 23). *Curso Java - 5: Arreglos y Matrices (Arrays)* [Video]. YouTube. <https://youtu.be/KZwMRL2q6O8>
- Deitel, P. (2016). *Cómo programar en Java*. Pearson.
- Programación ATS. (2017, abril 2). *Programación en Java || Arreglos || Definición de Arreglos* [Video]. YouTube. <https://youtu.be/AZc3hxMxl54?t=11>
- Programación ATS. (2017, julio 19). *70. Programación en Java || POO || Miembros estáticos de una clase* [Video]. YouTube. <https://youtu.be/aiBzXQmtzrA>
- Sánchez Allende, J. (2005). *Java 2: iniciación y referencia*. McGraw-Hill. <https://elibro.net/es/ereader/ues/50290?page=156>
- Sznajdleder, P. (2020). *Java a fondo*. Alfaomega.

NORMAS TÉCNICAS APLICABLES

No aplica



ANEXOS

No aplica



UES

Universidad Estatal de Sonora
La Fuerza del Saber Estimulará mi Espíritu