

## Proyecto práctico programado

**Temática:** Proyecto práctico: programa de computadora

**Valor porcentual:** 50%

### Instrucciones:

- El proyecto práctico programado consiste en la creación de un programa de computadora en el cual se soluciona una problemática de mediana complejidad. Los estudiantes deberán utilizar el paradigma imperativo, algoritmos con estructuras secuenciales, condicionales e iterativas y abstracción procedimental implementadas en el lenguaje de programación de alto nivel Java.
- La entrega final del proyecto debe ser entregado en formato .zip (carpeta comprimida que contiene la documentación solicitada), en el espacio indicado en Moodle.
- Debe seguir el siguiente formato de nombre para subir el archivo en Moodle:
  - NombreGrupoEntregaFinal

## Proyecto: “Aventura en la Isla del Tesoro”

### Descripción del juego:

El juego se basa en texto y simula una aventura en la isla. El jugador interactúa con el juego ingresando comandos y tomando decisiones que afectan el curso de la historia. El objetivo final es encontrar el tesoro escondido.

### Historia:

*“La leyenda cuenta que el famoso pirata Bytero de Nibble escondió su tesoro más preciado en una isla remota y misteriosa. Muchos han intentado encontrarlo, pero ninguno ha regresado. Se dice que la isla está llena de peligros, trampas y criaturas extrañas. En la U Cenfotec, un grupo de valientes estudiantes de programación ha decidido aceptar el desafío. Motivados por la sed de aventura y el deseo de poner a prueba sus habilidades, este equipo se embarcará en una peligrosa misión para encontrar el legendario tesoro de Barbanegra. ¿Serán capaces de desentrañar los misterios de la isla, superar los desafíos utilizando su ingenio y conocimientos de programación, y regresar a Cenfotec con el tesoro?”*

## Entregable 1: “El mapa y la brújula” (Semana 6)

### Objetivo:

Construir la base del juego utilizando estructuras secuenciales.

**Valor: 10%**

### Elementos del juego:

- Crear un mapa de la isla con al menos 5 ubicaciones (playa, cueva, jungla, ruinas, montaña).
- Cada ubicación debe tener una descripción detallada que se muestre al jugador.
- El jugador puede moverse entre las ubicaciones utilizando comandos como “ir al norte”, “ir al sur”, etc.
- Implementar un sistema básico de inventario con capacidad para 3 objetos. El jugador puede recoger objetos como una “pala”, una “antorcha” o un “mapa antiguo”.

### Ejemplo de interacción:

*Estás en la playa. Ves un cofre viejo semienterrado en la arena.*

*¿Qué deseas hacer?*

1. Examinar cofre
2. Ir al norte

### Conceptos aprendidos:

Variables.

Tipos de datos.

Operaciones de entrada y salida.

Estructuras secuenciales.

### Entregables:

1. Programa .java
2. Diagrama general y explicativo del programa a desarrollar
3. Tablas de datos de entrada
4. Tabla de datos de salida
5. Tabla de datos intermedios
6. Tabla de datos establecidos o constantes (opcional)

7. Los diagramas de flujo que facilitan la solución del problema
8. Dos casos de prueba que muestran el funcionamiento del programa

Esta actividad será evaluada por la siguiente rúbrica:

Criterio	Deficiente (1 punto)	Regular (2 puntos)	Bueno (3 puntos)	Excelente (4 puntos)
Diagrama general del problema	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma insuficiente	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma básica y presenta varios errores	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma regular y presenta algún error	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma adecuada y amplia
Diagrama explicativo	Confecciona el diagrama explicativo de la solución del problema de manera escasa	Confecciona el diagrama explicativo de la solución del problema de manera básica	Confecciona el diagrama explicativo de la solución del problema de manera regular	Confecciona el diagrama explicativo de la solución del problema de manera adecuada
Diagramas de flujo	Elabora el diagrama de flujo de la solución del problema de manera escasa y con múltiples errores	Elabora el diagrama de flujo de la solución del problema de manera básica y con varios errores	Elabora el diagrama de flujo de la solución del problema de manera regular y con algún error	Elabora el diagrama de flujo de la solución del problema de manera suficiente y sin errores
Casos de prueba	Contiene uno o dos casos presentados con múltiples errores	Contiene solo un caso de prueba completo. El otro caso de prueba está ausente o con errores	Contiene dos casos de prueba, pero presentan algún error.	Contiene dos casos de prueba completos
Programa de computadora	Implementa el programa de manera incorrecta y calcula lo que se le pide erróneamente.	Implementa el programa de manera básica y calcula lo que se pide con varios errores.	Implementa el programa de manera regular y calcula lo que se pide con algún error	Implementa el programa de manera clara y calcula lo que se pide sin errores

Estructuras de flujo	Utiliza las estructuras de flujo en el programa de manera inadecuada e ilógica	Utiliza las estructuras de flujo en el programa de manera básica y con poca lógica	Utiliza las estructuras de flujo en el programa de manera regular y lógica	Utiliza las estructuras de flujo en el programa de manera adecuada y lógica
Abstracción procedimental	Utiliza los arreglos, matrices y funciones de manera inadecuada	Utiliza los arreglos, matrices y funciones de manera básica	Utiliza los arreglos, matrices y funciones de manera regular	Utiliza los arreglos, matrices y funciones de manera suficiente
Funcionamiento del programa	Verifica la funcionalidad del programa por medio de casos de prueba de manera inapropiada y presenta múltiples errores	Verifica la funcionalidad del programa por medio de casos de prueba de manera elemental y presenta varios errores	Verifica la funcionalidad del programa por medio de casos de prueba de manera regular y presenta algún error	Verifica la funcionalidad del programa por medio de casos de prueba de manera apropiada y no presenta errores
Buenas prácticas	Utiliza las buenas prácticas en la implementación del programa de manera inadecuada	Utiliza las buenas prácticas en la implementación del programa de manera básica	Utiliza las buenas prácticas en la implementación del programa de manera regular	Utiliza las buenas prácticas en la implementación del programa de manera apropiada
Explicación del programa computacional	Realiza una explicación del programa de forma escueta y contesta de forma inadecuada las preguntas hechas por el docente	Realiza una explicación del programa de forma básica y contesta de forma básica varias de las preguntas hechas por el docente	Realiza una explicación del programa de forma regular y contesta alguna de las preguntas hechas por el docente de forma errónea	Realiza una explicación del programa de forma adecuada y contesta adecuadamente las preguntas hechas por el docente
Retroalimentación del docente	Incorpora la retroalimentación brindada por el docente de forma incoherente y escasa	Incorpora la retroalimentación brindada por el docente de manera básica	Incorpora la retroalimentación brindada por el docente de manera incompleta	Incorpora la retroalimentación brindada por el docente de manera clara y completa

## Entregable 2: “Desafíos y pistas” (Semana 10)

**Objetivo:**

Añadir elementos interactivos al juego utilizando estructuras condicionales.

**Valor: 10%****Elementos del juego:**

- Incorporar al menos 2 acertijos o desafíos en diferentes ubicaciones. Por ejemplo:
  - En la cueva, el jugador debe resolver un acertijo para abrir una puerta secreta.
  - En las ruinas, el jugador debe encontrar la combinación correcta para desactivar una trampa.
- Utilizar estructuras condicionales para evaluar las respuestas del jugador y determinar si avanza o no.
- Añadir pistas que ayuden al jugador a resolver los acertijos. Las pistas pueden estar escondidas en otras ubicaciones o ser obtenidas al interactuar con ciertos objetos.
- Implementar un sistema de puntuación que aumente al resolver acertijos o encontrar objetos importantes.

**Ejemplo de interacción:**

*Estás en la cueva. Frente a ti hay una puerta con un dibujo extraño. Parece un acertijo...*

*¿Qué deseas hacer?*

*> Examinar dibujo*

*> Usar antorcha*

*> Ir al este*

**Conceptos aprendidos:**

- Operadores relacionales y lógicos.
- Estructuras condicionales (if-else).

**Entregables:**

1. Programa .java
2. Diagrama general y explicativo del programa a desarrollar
3. Tablas de datos de entrada
4. Tabla de datos de salida

5. Tabla de datos intermedios
6. Tabla de datos establecidos o constantes (opcional)
7. Los diagramas de flujo que facilitan la solución del problema
8. Dos casos de prueba que muestran el funcionamiento del programa

Esta actividad será evaluada por la siguiente rúbrica:

Criterio	Deficiente (1 punto)	Regular (2 puntos)	Bueno (3 puntos)	Excelente (4 puntos)
Diagrama general del problema	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma insuficiente	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma básica y presenta varios errores	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma regular y presenta algún error	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma adecuada y amplia
Diagrama explicativo	Confecciona el diagrama explicativo de la solución del problema de manera escasa	Confecciona el diagrama explicativo de la solución del problema de manera básica	Confecciona el diagrama explicativo de la solución del problema de manera regular	Confecciona el diagrama explicativo de la solución del problema de manera adecuada
Diagramas de flujo	Elabora el diagrama de flujo de la solución del problema de manera escasa y con múltiples errores	Elabora el diagrama de flujo de la solución del problema de manera básica y con varios errores	Elabora el diagrama de flujo de la solución del problema de manera regular y con algún error	Elabora el diagrama de flujo de la solución del problema de manera suficiente y sin errores
Casos de prueba	Contiene uno o dos casos presentados con múltiples errores	Contiene solo un caso de prueba completo. El otro caso de prueba está ausente o con errores	Contiene dos casos de prueba, pero presentan algún error.	Contiene dos casos de prueba completos
Programa de computadora	Implementa el programa de manera incorrecta y calcula lo que se le pide erróneamente.	Implementa el programa de manera básica y calcula lo que se pide con varios errores.	Implementa el programa de manera regular y calcula lo que se pide con algún error	Implementa el programa de manera clara y calcula lo que se pide sin errores

Estructuras de flujo	Utiliza las estructuras de flujo en el programa de manera inadecuada e ilógica	Utiliza las estructuras de flujo en el programa de manera básica y con poca lógica	Utiliza las estructuras de flujo en el programa de manera regular y lógica	Utiliza las estructuras de flujo en el programa de manera adecuada y lógica
Abstracción procedimental	Utiliza los arreglos, matrices y funciones de manera inadecuada	Utiliza los arreglos, matrices y funciones de manera básica	Utiliza los arreglos, matrices y funciones de manera regular	Utiliza los arreglos, matrices y funciones de manera suficiente
Funcionamiento del programa	Verifica la funcionalidad del programa por medio de casos de prueba de manera inapropiada y presenta múltiples errores	Verifica la funcionalidad del programa por medio de casos de prueba de manera elemental y presenta varios errores	Verifica la funcionalidad del programa por medio de casos de prueba de manera regular y presenta algún error	Verifica la funcionalidad del programa por medio de casos de prueba de manera apropiada y no presenta errores
Buenas prácticas	Utiliza las buenas prácticas en la implementación del programa de manera inadecuada	Utiliza las buenas prácticas en la implementación del programa de manera básica	Utiliza las buenas prácticas en la implementación del programa de manera regular	Utiliza las buenas prácticas en la implementación del programa de manera apropiada
Explicación del programa computacional	Realiza una explicación del programa de forma escueta y contesta de forma inadecuada las preguntas hechas por el docente	Realiza una explicación del programa de forma básica y contesta de forma básica varias de las preguntas hechas por el docente	Realiza una explicación del programa de forma regular y contesta alguna de las preguntas hechas por el docente de forma errónea	Realiza una explicación del programa de forma adecuada y contesta adecuadamente las preguntas hechas por el docente
Retroalimentación del docente	Incorpora la retroalimentación brindada por el docente de forma incoherente y escasa	Incorpora la retroalimentación brindada por el docente de manera básica	Incorpora la retroalimentación brindada por el docente de manera incompleta	Incorpora la retroalimentación brindada por el docente de manera clara y completa

## Entregable 3: “El tesoro y la victoria” (Semana 14)

### Objetivo:

Completar el juego utilizando estructuras iterativas y funciones.

### Elementos del juego:

- Esconder el tesoro en una ubicación secreta, como en la cima de la montaña.
- Crear un guardián del tesoro, como un “esqueleto pirata” o un “pulpo gigante”. El jugador debe vencer al guardián para obtener el tesoro. El combate puede ser basado en texto, donde el jugador elige acciones como “atacar” o “defender”.
- Utilizar estructuras iterativas para controlar el flujo del juego, como:
- Un bucle que se repite hasta que el jugador encuentra el tesoro o pierde todas sus vidas.
- Un menú que permite al jugador elegir entre diferentes acciones en cada ubicación.
- Crear funciones para organizar el código, como una función para mostrar el inventario, una función para el combate, etc.
- Implementar diferentes finales según las acciones del jugador. Por ejemplo:
- Encontrar el tesoro y escapar de la isla.
- Ser derrotado por el guardián del tesoro.
- Quedar atrapado en una trampa mortal.

### Ejemplo de interacción:

*Has llegado a la cima de la montaña. Frente a ti está el tesoro de Barbanegra. Pero un esqueleto pirata aparece para protegerlo!*

*¿Qué deseas hacer?*

*> Atacar*

*> Defender*

*> Huir*

### Conceptos aprendidos:

- Estructuras iterativas (for, while).
- Funciones.
- Modularidad.



**Consideraciones adicionales:**

- El código del juego debe estar bien documentado y seguir las buenas prácticas de programación aprendidas en clase.
- Se puede fomentar la creatividad de los estudiantes permitiéndoles personalizar el juego con su propia historia, personajes y desafíos.

**Entregables:**

1. Programa .java
2. Diagrama general y explicativo del programa a desarrollar
3. Tablas de datos de entrada
4. Tabla de datos de salida
5. Tabla de datos intermedios
6. Tabla de datos establecidos o constantes (opcional)
7. Los diagramas de flujo que facilitan la solución del problema
8. Dos casos de prueba que muestran el funcionamiento del programa

Esta actividad será evaluada por la siguiente rúbrica:

Criterio	Deficiente (1 punto)	Regular (2 puntos)	Bueno (3 puntos)	Excelente (4 puntos)
Diagrama general del problema	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma insuficiente	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma básica y presenta varios errores	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma regular y presenta algún error	Elabora el diagrama general, las tablas de datos de entrada, salida, intermedios y constantes de forma adecuada y amplia
Diagrama explicativo	Confecciona el diagrama explicativo de la solución del problema de manera escasa	Confecciona el diagrama explicativo de la solución del problema de manera básica	Confecciona el diagrama explicativo de la solución del problema de manera regular	Confecciona el diagrama explicativo de la solución del problema de manera adecuada

Diagramas de flujo	Elabora el diagrama de flujo de la solución del problema de manera escasa y con múltiples errores	Elabora el diagrama de flujo de la solución del problema de manera básica y con varios errores	Elabora el diagrama de flujo de la solución del problema de manera regular y con algún error	Elabora el diagrama de flujo de la solución del problema de manera suficiente y sin errores
Casos de prueba	Contiene uno o dos casos presentados con múltiples errores	Contiene solo un caso de prueba completo. El otro caso de prueba está ausente o con errores	Contiene dos casos de prueba, pero presentan algún error.	Contiene dos casos de prueba completos
Programa de computadora	Implementa el programa de manera incorrecta y calcula lo que se le pide erróneamente.	Implementa el programa de manera básica y calcula lo que se pide con varios errores.	Implementa el programa de manera regular y calcula lo que se pide con algún error	Implementa el programa de manera clara y calcula lo que se pide sin errores
Estructuras de flujo	Utiliza las estructuras de flujo en el programa de manera inadecuada e ilógica	Utiliza las estructuras de flujo en el programa de manera básica y con poca lógica	Utiliza las estructuras de flujo en el programa de manera regular y lógica	Utiliza las estructuras de flujo en el programa de manera adecuada y lógica
Abstracción procedimental	Utiliza los arreglos, matrices y funciones de manera inadecuada	Utiliza los arreglos, matrices y funciones de manera básica	Utiliza los arreglos, matrices y funciones de manera regular	Utiliza los arreglos, matrices y funciones de manera suficiente
Funcionamiento del programa	Verifica la funcionalidad del programa por medio de casos de prueba de manera inapropiada y presenta múltiples errores	Verifica la funcionalidad del programa por medio de casos de prueba de manera elemental y presenta varios errores	Verifica la funcionalidad del programa por medio de casos de prueba de manera regular y presenta algún error	Verifica la funcionalidad del programa por medio de casos de prueba de manera apropiada y no presenta errores
Buenas prácticas	Utiliza las buenas prácticas en la implementación del programa de manera inadecuada	Utiliza las buenas prácticas en la implementación del programa de manera básica	Utiliza las buenas prácticas en la implementación del programa de manera regular	Utiliza las buenas prácticas en la implementación del programa de manera apropiada

Explicación del programa computacional	Realiza una explicación del programa de forma escueta y contesta de forma inadecuada las preguntas hechas por el docente	Realiza una explicación del programa de forma básica y contesta de forma básica varias de las preguntas hechas por el docente	Realiza una explicación del programa de forma regular y contesta alguna de las preguntas hechas por el docente de forma errónea	Realiza una explicación del programa de forma adecuada y contesta adecuadamente las preguntas hechas por el docente
Retroalimentación del docente	Incorpora la retroalimentación brindada por el docente de forma incoherente y escasa	Incorpora la retroalimentación brindada por el docente de manera básica	Incorpora la retroalimentación brindada por el docente de manera incompleta	Incorpora la retroalimentación brindada por el docente de manera clara y completa