

Segundo Proyecto Programado

Batalla Naval

Curso: Taller de Programación

Fecha Inicio: martes 02 de abril de 2024 12:30pm

Fecha de entrega: martes 23 de abril 2024, 12:30pm

Valor Porcentual: 20%

Objetivo

Propiciar que los estudiantes pongan en práctica la programación con matrices como estructuras de datos y el almacenamiento y lectura de archivos mediante el desarrollo de un juego de batalla naval.

Descripción General

Batalla naval o "Battleship" es un juego de estrategia en el que dos jugadores compiten para hundir los barcos del oponente antes de que este hunda los suyos. Cada jugador tiene un tablero separado en el que coloca sus propios barcos sin que el oponente pueda ver su disposición. Los jugadores se turnan para intentar adivinar la ubicación de los barcos enemigos y lanzar ataques. El objetivo es adivinar correctamente las coordenadas donde se encuentran los barcos del oponente y hundirlos antes de que el oponente hunda los propios.

Jugadores: Para el desarrollo del juego se deben registrar los dos jugadores con el nombre y un nickname.

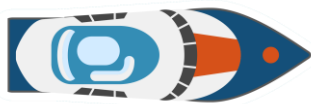
Naves: Cada jugador dispondrá de 3 tipos de naves distintas:

- **Destructor:**



- *Tamaño:* 1 celda
- *Descripción:* El destructor es el barco más pequeño y ágil de la flota. Debido a su tamaño compacto, puede maniobrar fácilmente entre las aguas enemigas.
- *Estrategia:* Debido a su tamaño reducido, el destructor puede ocultarse en áreas más pequeñas del tablero y ser más difícil de encontrar para el oponente.
- *Puntos:* 1

- **Crucero:**



- *Tamaño:* 2 celdas
- *Descripción:* El crucero es un barco de tamaño mediano que combina la capacidad de maniobra del destructor con una mayor resistencia y capacidad de ataque.
- *Estrategia:* El crucero puede ser utilizado tanto para patrullar áreas extensas del tablero como para emboscar a los barcos enemigos con mayor eficacia debido a su tamaño y versatilidad.
- *Puntos:* 2

- **Acorazado:**



- *Tamaño:* 3 celdas
- *Descripción:* El acorazado es el buque más grande y resistente de la flota, con una formidable capacidad de combate y defensa.
- *Estrategia:* El acorazado puede usarse como focal de la flota, atrayendo la atención del enemigo mientras los barcos más pequeños atacan sorpresa desde posiciones más ocultas. Su tamaño imponente puede intimidar al oponente y obligarlo a desviar recursos para neutralizarlo.
- *Puntos:* 3.

Tablero: Cada jugador dispone de dos tableros de las mismas dimensiones, uno representa la ubicación de sus naves y el otro representa la zona de tiro a la que está lanzado a su enemigo. Al iniciar el juego se debe definir las dimensiones de ambos tableros juntos, las que

no pueden ser menores a 10 filas X 20 columnas, y la cantidad de columnas siempre debe ser un número par, de modo que la mitad de las columnas de la izquierda pertenezcan al área de juego del primer jugador y las de la derecha del segundo jugador. Al desplegar los barcos en el juego se deberán mostrar imágenes que reflejen la dirección de que se desplacen y, en el caso de las naves que abarquen múltiples celdas, se seccionarán en múltiples imágenes para representar embarcaciones grandes.

Colocación de las naves: Todos los botes serán colocados en el tablero por cada jugador (no se pueden colocar superpuestos los botes). Cada jugador tendrá a disposición la colocación 4 cruceros, 2 acorazados y 6 destructores. Al colocarlo en el tablero se deberá establecer la dirección horizontal (izquierda o derecha) y vertical (arriba o abajo).

Movimiento de las naves: Tras cada ronda del juego (cuando ambos jugadores hayan participado) las naves deben avanzar según la dirección en el plano 2D donde se encuentran. Si las naves están posicionadas horizontalmente estas podrán desplazarse de izquierda a derecha, si están posicionadas verticalmente estas pueden desplazarse de arriba a abajo. Debe establecerse el sentido en el que se desplaza cada nave, si estas chocan con otra nave o con los bordes de la matriz no deben avanzar y por el contrario estas cambian de sentido, es decir si se desplazan de derecha a izquierda y chocan en el borde izquierdo, la nave se queda estática en su posición, pero en la próxima ronda esta misma nave deberá desplazarse de izquierda a derecha. Por ronda las naves de tipo “crucero” y “acorazado” solo avanzan una posición, pero las “destructor” avanzan 1 o dos espacios según las otras naves o los bordes lo permitan.

Desarrollo del Juego: Cada jugador de manera alterna debe ir jugando, indicando las posiciones a donde apunta su disparador directamente seleccionando la celda en la matriz del juego, si logra pegar una de las posiciones de la nave, se debe encender un apuntador en rojo del tablero de la zona de tiro (oponente), si el barco es de tipo destructor al tener un ancho de una celda, este se destruye de inmediato y los puntos se acumulan para el usuario respectivo, si las naves son de tipo acorazado y crucero, cuyo tamaños de celda son de 3 y 2 celdas respectivamente, si el tiro tocó alguna de estas celdas, el barco no puede seguir su curso y los puntos se acumularán únicamente cuando el jugador en turno haya destruido el barco por completo (todas las celdas), sólo se hace un disparo por turno.

Si es el turno del jugador 1 se deberán hacer visibles las naves del este y se ocultarán las del jugador 2. Si es el turno del jugador 2 se deberán hacer visibles las naves del este y se ocultarán las del jugador 1, solo se mantendrán activas las celdas en rojo de los barcos semidestruidos.

En todo momento se deben mostrar el recuento de barcos disponibles por cada jugador y el puntaje de actual de los jugadores.

Fin del Juego: Existen dos posibilidades de fin de juego

- **Ganador:** quien destruya primero todas las naves de su oponente será el vencedor (al participar por turnos, en caso de que el participante que comenzó la partida hunda en su última jugada el último barco de su oponente que quedaba a flote, el otro participante tiene derecho a una última posibilidad para alcanzar el empate, a un último disparo que también le permita terminar de hundir la flota contraria, lo que supondría un empate);
- **Empate:** si bien lo habitual es continuar hasta que haya un ganador, el empate puede darse cuando definan terminar la partida y cada jugador disparó igual de tiros y ambos jugadores han acertado en igual número de casillas contrarias.

Guardar el juego: la partida actual con el puntaje nombre y nickname de jugadores, y el estado, las posiciones y direcciones de naves restantes, deberán ser almacenados en archivos para posteriormente continuar las partidas almacenadas.

Para el desarrollo del juego se le solicita:

1. **Creación del tablero de juego:** El programa crea una matriz para representar el tablero de juego, donde cada celda de la matriz representa una posición en el mar.
2. **Colocación de barcos:** Permitiría a los jugadores colocar sus barcos en el tablero antes de comenzar el juego, verificando si las posiciones son válidas y si hay colisiones con otros barcos
3. **Guardado y carga de partidas:** El programa permitiría guardar el estado actual del juego en un archivo, incluyendo la disposición de los barcos y cualquier otro estado relevante. También podría cargar partidas guardadas previamente desde archivos para reanudar el juego en otro momento, para esto debe elegirse un nombre de partida.
4. **Ataque y respuesta:** Los jugadores pueden seleccionar una posición en el tablero para lanzar un ataque. El programa verificaría si la posición seleccionada contiene un barco enemigo y actualizaría el estado del juego en consecuencia. La respuesta del oponente también se registrará y mostrará (Es decir se debe alertar donde disparó el oponente en su última partida).
5. **Verificación de ganador:** Después de cada turno de ataque, el programa verificaría si alguno de los jugadores ha hundido todos los barcos del oponente. En caso afirmativo, el programa declararía al jugador correspondiente como ganador y finalizaría el juego.

6. **Interfaz de usuario:** Interfaz gráfica para que los jugadores interactúen con el juego. Esto incluiría mostrar el estado actual del tablero, mensajes de estado y opciones de entrada para colocar barcos y lanzar ataques.
7. **Validación de entrada:** El programa validaría las entradas de los usuarios para garantizar que sean coordenadas válidas dentro de los límites del tablero y que cumplan con el formato esperado.
8. **Mensajes y notificaciones:** Proporcionaría mensajes claros y notificaciones para informar a los jugadores sobre el progreso del juego, como "¡Has golpeado un barco enemigo!" o "¡Tu barco ha sido alcanzado!".
9. **Archivos:** el programa puede almacenar información sobre el estado del juego, como la ubicación de los barcos y los ataques realizados, lo que permite una experiencia de juego persistente y la posibilidad de reanudar partidas en el futuro.

Evaluación

| Rubro | Valor porcentual |
|--|------------------|
| Registro de jugadores | 5% |
| Creación de matriz del juego | 5% |
| Posicionamiento de flota de cada jugador | 5% |
| Desarrollo del juego | 40% |
| Movimiento de botes | 10% |
| Guardado del juego | 10% |
| Documentación interna y notación | 5% |
| Uso de control de versiones | 5% |
| Documentación externa | 15% |

Aspectos Administrativos

1. Todo el código debe ser escrito en el lenguaje Python.
2. El proyecto se considera como una entidad única. Aunque la evaluación se divida por secciones, la falta de completitud en una o más áreas puede afectar la funcionalidad general del sistema y, por ende, influir en la nota final.
3. El proyecto puede ser desarrollado por un máximo de dos personas. Sin embargo, se aplican las siguientes reglas para los equipos de trabajo:

1. En caso de que una pareja no pueda continuar trabajando junta por cualquier razón, ambos deben continuar utilizando el código fuente que han desarrollado hasta el momento. No está permitido que se unan a otro compañero ni que utilicen el código fuente de otro compañero.
2. Aunque el trabajo sea realizado en equipo, la evaluación y asignación de la nota se hará de forma individual para cada estudiante. Esto se aplica en situaciones donde el docente determine que uno de los estudiantes tuvo poca o nula participación en el desarrollo del proyecto, según las preguntas planteadas durante la defensa del proyecto.
4. En caso de fraude por parte de un estudiante, ya sea provocado o consentido, el proyecto será automáticamente anulado para todos los involucrados. Esto se encuentra en concordancia con el Artículo 75 del Reglamento del Régimen Enseñanza-Aprendizaje del Instituto Tecnológico de Costa Rica, disponible en el siguiente enlace: <https://www.tec.ac.cr/reglamentos/reglamento-regimen-ensenanza-aprendizaje-tecnologico-cost Rica-sus-reformas>.
5. Todos los archivos .py y la documentación deben ser subidos a la plataforma oficial utilizada en el curso. El docente copiará y pegará el código fuente en el intérprete para comprobar su funcionamiento. No se realizarán modificaciones al código fuente por parte del docente. Por lo tanto, es responsabilidad del estudiante asegurarse de que el código funcione correctamente.
6. El proyecto tiene una duración de 22 días (3 semanas) a partir de la fecha de entrega proporcionada por el profesor. No se aceptarán proyectos entregados después de la fecha límite establecida. Cualquier excepción a esta regla resultará en una penalización en la nota.
7. Debe hacerse uso de algún sistema de control de versiones que los estudiantes definan y asignar permisos de acceso al docente para su respectivo seguimiento.
8. Cada fuente y referencia bibliográfica utilizada deberá documentarse según lo indica el siguiente apartado, de lo contrario se considerará como fraude. Se recomienda hacer uso de las bases de datos digitales de la biblioteca en el sitio Web <https://www.tec.ac.cr/sistema-bibliotecas-tec>
9. Se debe documentar todo el código generado utilizando el estándar de documentación docString.

Incluya la siguiente tabla en la documentación externa describiendo el grado de participación de cada integrante del grupo.

| Rubro | Nombre estudiante 1 | Nombre estudiante 2 |
|--|---------------------|---------------------|
| Registro de jugadores | | |
| Creación de matriz del juego | | |
| Posicionamiento de flota de cada jugador | | |
| Desarrollo del juego | | |
| Movimiento de botes | | |
| Guardado del juego | | |
| Documentación interna y notación | | |
| Uso de control de versiones | | |
| Documentación externa | | |

Estándares de codificación para proyectos programados: Es obligación aplicar en el proyecto alguna notación de programación reconocida para la definición de nombres de funciones, variables y otros elementos del lenguaje que se maneje, por ejemplo:

- Notación CamelCase:
 - LowerCamelCase: Cada variable deberá definirse con su primer palabra en minúscula y las siguientes con su primer letra en mayúsculas. Ejemplos: primerNombre, numeroPuestos, correoPersonal.
 - UpperCamelCase: Cada variable deberá definirse con la primer letra de sus palabras en mayúscula. Ejemplos: PrimerNombre, NumeroPuestos, CorreoPersonal.
- Notación SnakeCase: Cada una de las palabras estarán unidas con un guion bajo (_). Ejemplos: primer_nombre, numero_puestos, correo_personal.
- Notación Húngara: Es similar a la notación CamelCase, pero agrega un prefijo antes del nombre. Ejemplo: intContador, chkPuesto, frmPrincipal

Documentación Externa

La definición de cada proyecto programado para algún curso de la Unidad de Ingeniería en Computación deberá constituir un documento formal, donde el docente establezca claramente las pautas a seguir para su ejecución y evaluación.

En particular cada proyecto debe exigir una documentación escrita que cumpla con los requisitos mínimos indicados en esta guía. Cada profesor será el responsable de adjuntar los siguientes apartados en la definición del proyecto a asignar. Esto no implica que el mismo pueda adicionar requerimientos extra basados en su juicio experto.

A continuación, se describen los apartados básicos que el estudiante deberá presentar en las documentaciones de sus proyectos programados.

Portada

[Se deberá seguir el estándar siguiente]

| |
|--|
| <p><i>Instituto Tecnológico de Costa Rica</i> <i>Unidad de Computación</i></p> <p><i>[“Título del trabajo”]</i></p> <p><i>[Nombre del estudiante]</i></p> <p><i>Sede San Carlos</i> <i>[Fecha]</i></p> |
|--|

Introducción

[Se realiza una breve presentación de la temática a tratar en el proyecto o asignación.

“Debe” responder directamente al porqué se realiza el proyecto o asignación. Resulta útil incluir información sobre las últimas tendencias o enfoques aplicados en el área al que pertenece la asignatura.

Por otra parte, debe evitarse a toda costa:

1. *Redundar nuevamente en los puntos indicados por parte del docente en el proyecto*

2. *Análisis de tipo filosófico, que introduzcan aspectos que no se relacionan directamente con el proyecto, por ejemplo:*

“Desde el inicio, el ser humano ha emprendido una búsqueda incansable en pos de convertirse...”

“En la actualidad las Tecnologías de la Información juegan un papel importante en cualquier ámbito de la sociedad...”

Se pueden presentar ciertos aspectos que evidencian la estructura del informe o artículo, sin entrar necesariamente en aspectos de fondo]

Análisis del problema

[El análisis del problema constituye una herramienta sumamente útil para formar en los alumnos la cultura de la buena comunicación en el ámbito de la Ingeniería de Software.

Resulta conveniente analizar los puntos a tratar en la evaluación, considerando estos como requerimientos de un proyecto de software o como las preguntas o situaciones a resolver en un proceso de indagatoria (investigación).

No obstante, se deben considerar mecanismos que permitan llevar a cabo una descripción eficiente de la situación y las metodologías o medios necesarios para brindar una posible solución o abordaje al problema.

Para ello, se pueden considerar: diagramas de conceptos, técnicas, herramientas, o ejemplos prácticos de situaciones en donde se visualice la problemática o situación planteada, evitando a toda costa la utilización de descripciones o prosa confusa y poco relevante sobre el punto a tratar. Un ejemplo incorrecto sería:

“1. Implementar un analizador de texto, para determinar el idioma. Aquí los alumnos deben indagar sobre cómo analizar texto para determinar el idioma del mismo. Jorge Alfaro Velasco es el encargado de analizar este problema, no obstante, Lorena Valerio también ayudará en la solución, ya que como dice el refrán dos cabezas piensan mejor que una...”]

Solución del problema

[Por otra parte, en contraposición con el análisis del problema, este apartado representa el planteamiento o enfoque para afrontar la temática, situación o problema asignado.

Es muy importante comunicar de manera eficiente la solución adoptada a partir del análisis realizado.

Uno de los mejores ejemplos lo constituyen las páginas o sitios WEB enfocados en la solución de problemas o situaciones en el ámbito del software. Por ejemplo: <http://www.codeproject.com/> o www.stackoverflow.com se utilizan en forma frecuente para obtener o poner a disposición información relacionada con un problema en particular. Los problemas se abordan a través de los pasos que permiten llevar a cabo su solución, empleando recursos como:

- 1. Etapas de la solución*
- 2. Diagramas e imágenes de la solución planteada*
- 3. Código e incluso pseudocódigo de la posible solución*

Debe evitarse la descripción de la solución final implementada para el proyecto, en otras palabras, un resumen escueto de las labores realizadas, y que por lo general se agregan una vez realizado el proyecto o poco antes de entregarlo, de manera tal que este apartado se degrada o convierte en una bitácora de las labores realizadas. Un ejemplo incorrecto sería:

“Los alumnos se reunieron e implementaron dos métodos, uno principal con los pasos necesarios para llevar a cabo las operaciones aritméticas y otro que se encarga de imprimir los resultados en consola...”

“Método ObtenerDatos() Se encarga de obtener los datos suministrados por el usuario y los transfiere a la capa de negocios...”]

Instalación:

Se deben indicar las instrucciones detalladas para instalar el software y la Configuración inicial.

Pruebas:

Estrategia de pruebas utilizada y Resultados de las pruebas realizadas a su aplicación para comprobar que satisface lo solicitado por el profesor.

Análisis de resultados

[En este apartado se debe buscar el mayor acercamiento entre los alumnos y la forma en que deben comunicarse los avances y resultados de los proyectos a nivel profesional, por lo tanto, resulta esencial fomentar la visión de este apartado como un análisis del estado de los requerimientos de un proyecto.

Debe ser claro y conciso a la hora de expresar estos resultados, sin llevar a cabo un análisis exhaustivo o comentarios que de alguna manera desvíen al lector (cliente, usuario) del objetivo principal: Determinar el estado actual del proyecto.

Por ende, se recomienda contar con alguna tabla o formato que permita determinar de forma rápida las labores realizadas y visualizar un panorama general.

A continuación, un ejemplo del formato para expresar los resultados:

| Tarea/Requerimiento | Estado | Observaciones |
|----------------------------|---------------|--|
| Gestión de usuarios | Completo | |
| Ordenamiento de lista | Incompleto | El algoritmo no realiza el ordenamiento en ciertos casos |

Los datos por incluir en este análisis deben relacionarse estrictamente con el proyecto, no deben girar en torno a percepciones u opiniones personales de los involucrados]

Conclusiones

[Deben dirigirse estrictamente a los resultados obtenidos en el proyecto. Por ejemplo:

- 1. Emitir un criterio respecto al rendimiento de un proceso implementado en contraposición con uno alternativo.*
- 2. Pertinencia de la aplicación de un enfoque de programación en algún ámbito en particular de acuerdo con la experiencia o resultados del proyecto.*
- 3. Resultados concretos del proyecto, por ejemplo: “Se logró determinar la eficiencia del proceso X con respecto al Y, en los ámbitos que implican...”*

No deben incluirse aspectos o criterios personales, por ejemplo:

“Al concluir este proyecto aprendimos sobre el funcionamiento de las estructuras de datos como: árboles y grafos, y la importancia de estos...”

“El actual proyecto nos enseña sobre la importancia de los sistemas de gestión de bases de datos en el ámbito empresarial...”]

Recomendaciones

[Al igual que las conclusiones, se trata de un espacio en donde se pueden plantear:

- 1. Aspectos o ámbitos de interés para ampliar el alcance del proyecto.*
- 2. Áreas o campos de estudio que pueden complementar las temáticas analizadas.*
- 3. Ideas que sean de utilidad para interesados en el área en que se desenvuelve el proyecto.*

No se deben abordar temáticas o aspectos relacionados con la percepción de los integrantes, y que en general no aportan en absoluto al tema principal, por ejemplo:

“Se considera que el tiempo para realizar el proyecto fue...”

“Se recomienda que el profesor brinde una explicación sobre temas...”]

Consideraciones Generales

1. *Cada fuente y referencia bibliográfica utilizada deberá documentarse según esta guía lo indica, de lo contrario se considerará como fraude. Se recomienda hacer uso de las bases de datos digitales de la biblioteca en el sitio Web <https://www.tec.ac.cr/sistema-bibliotecas-tec>*

2. *Estándares de codificación para proyectos programados: Es obligación aplicar en el proyecto alguna notación de programación reconocida para la definición de nombres de funciones, variables y otros elementos del lenguaje que se maneje, por ejemplo:*

a. *Notación CamelCase:*

LowerCamelCase: Cada variable deberá definirse con su primer palabra en minúscula y las siguientes con su primer letra en mayúsculas.

Ejemplos: primerNombre, numeroPuestos, correoPersonal.

UpperCamelCase: Cada variable deberá definirse con la primera letra de sus palabras en mayúscula.

Ejemplos: PrimerNombre, NumeroPuestos, CorreoPersonal.

b. *Notación SnakeCase: Cada una de las palabras estarán unidas con un guion bajo (_)*

Ejemplos: primer_nombre, numero_puestos, correo_personal

c. *Notación Húngara: Es similar a la notación CamelCase, pero agrega un prefijo antes del nombre.*

Ejemplo: intContador, chkPuesto, frmPrincipal

Otras consideraciones

Para proyectos que involucren la vinculación con algún ente externo se deberá solicitar adicional:

- *Nombre de empresa, dirección y contacto.*
- *Descripción de la empresa, [Estructura Organizacional].*
- *Objetivo General y específicos del proyecto particular.*
- *Justificación del proyecto.*
- *Supuestos y Restricciones presentes en este proyecto.*
- *[Requerimientos “específicos” del sistema].*

Exposición

Proyecto de programación: la presentación oral

Objetivos de aprendizaje: Este ejercicio busca que los estudiantes sean capaces de:

1. Explicar con seguridad los pasos seguidos en el desarrollo de un proyecto de programación.
2. Justificar de manera sólida y objetiva las decisiones tomadas en el proceso de desarrollo de software o algoritmo.
3. Fortalecer las habilidades de comunicación oral en la defensa de un proyecto técnico.
4. Comprender la importancia de la comunicación oral en los espacios académicos y laborales.

Primera presentación: semana 7 (18 al 22 de marzo)

Cada pareja asistirá a la oficina del docente en la fecha y hora previamente acordadas. El objetivo consiste en exponer la propuesta del proyecto de programación (algoritmo, software). Si bien no se trata de una presentación formal de carácter expositivo (frente a un público amplio), cada pareja seguirá las pautas necesarias para que el docente -al ser el público meta- comprenda el trabajo realizado. Por ello, durante el transcurso de la presentación, se realizarán preguntas con el objetivo de cada pareja demuestre dominio y comprensión del tema. El tiempo estimado para esta presentación es de entre 20 y 30 minutos.

Resumen de las características de cada presentación

| | Segunda experiencia |
|------------------------------------|---|
| Semana | Semana 11 22 al 26 de abril |
| Objetivo | Presentar los resultados del proyecto de programación |
| Carácter de la presentación | Informativo |
| Tiempo para la presentación | 8 mínimo, 10 minutos máximo |
| Escenario | De pie frente al grupo en el aula |
| Público meta | Docente del curso y compañeros de clase |
| Número de integrantes | Parejas |
| Material de apoyo | Se envía una semana antes en formato ppt o pdf |
| Ronda de preguntas | Después de la fase de cierre |

Lineamientos generales para las presentaciones

Antes de la presentación

- Preparen y comprendan con suficiente tiempo los insumos teóricos-prácticos: caso contrario, planteen sus dudas con el docente a cargo.
- Planifiquen y practiquen su exposición con suficiente anticipación.
- Distribuyan las secciones de la presentación: por ejemplo, un estudiante realiza la introducción, otro el cierre, y entre ambos el desarrollo.
- Practiquen aspectos relacionados con la estilística oral: articulación, volumen, velocidad, tono. Prestar cuidado a la aparición de muletillas (se recomienda grabarse para revisar tales aspectos).
- Corrobores que cuentan con el material necesario para llevar a cabo la presentación.
- Guíense con la estructura de la presentación (ver punto Estructura de la presentación más abajo en este mismo documento)

Durante la presentación

- Asistan puntualmente.
- Lleven a mano el material de apoyo que consideren pertinente: el código y la documentación interna y externa.
- Cronometren el tiempo en que interviene cada participante.
- Tomen la iniciativa: no esperen que el docente les dé las pautas, al contrario, sean proactivos y dinámicos.
- Cuiden el lenguaje corporal: siéntense de manera erguida (en el caso de la primera experiencia), mantengan los hombros y espalda firmes, realicen contacto visual, muevan las manos suavemente, acorde con el mensaje. Cuando expongan de pie, mantengan una posición natural, procuren desplazarse, y establezcan contacto visual con toda la audiencia. Eviten cruzar los brazos, meter sus manos en las bolsas del pantalón o realizar movimientos constantes y repetitivos.
- Empleen un volumen alto, cuiden su articulación, velocidad y tono.
- Desarrollen ideas completas y coherentes y apóyense en conectores para pasar de un tema a otro. Por ejemplo, en primer lugar, vamos a explicar las características del problema.... Seguidamente, le mostramos la estructura del código...para finalizar...
- Eviten el uso excesivo de muletillas como “esteeeee”, “entonces”, “digamos”, “o sea”, entre otras.
- Empleen un vocabulario técnico y formal. Eviten palabras como “cosa”, “eso lo hicimos” y expresiones como “a nosotros nos tocó”, “esa parte no me tocó / correspondió” entre otras.
- En caso de haber consultado fuentes bibliográficas, procure reconocerlas debidamente durante su presentación.

Para finalizar la presentación

- Cierren de manera adecuada. Eviten frases como “eso sería”, “eso fue todo”, en su lugar, prefieran “Finalmente, agradecemos la atención”, “estamos atentos a cualquier pregunta o recomendación adicional”
- En caso de haber preguntas de parte del docente a cargo, respóndalas de manera respetuosa y cordial.
- Enlisten las observaciones realizadas por el docente a cargo, ya que se constituyen en una oportunidad de mejora para sus proyectos.

Estructura de la presentación

Apertura

- Saluden al público
- Empleen alguna estrategia de introducción: pregunta retórica, cita, dato curioso o estadístico, entre otras
- Comenten brevemente y de manera concreta los puntos (ideas principales) que abordarán en el desarrollo

Desarrollo

- Expliquen cada una de las ideas:
- Análisis del problema (¿cuál problema pretende resolver el software?)
- Solución del problema (expliquen con detalle en qué consiste)
- Estructura del código
- Justificación de las decisiones tomadas (argumenten, apoyándose en los conceptos estudiados en el curso, el porqué de cada decisión de diseño de software)
- Resultado (Expliquen cuál fue el resultado obtenido)

Cierre

- Resuman las ideas principales del desarrollo
- Reflexionen en torno a las limitaciones
- Refiéranse a los puntos de mejora
- Agradezcan la atención brindada
- Ofrezcan un espacio para preguntas y comentarios

Criterios de evaluación

Las presentaciones se evaluarán con el siguiente instrumento.

Nombre del estudiante/Integrantes del equipo:

Instrumento de evaluación por pareja (27 puntos)

| Criterio | 3 | 2 | 1 | 0 |
|--|--|---|---|---|
| Estructura | | | | |
| Fase de apertura | La apertura incluye todos los tres elementos sugeridos: saludo, estrategias y temas a tratar | La apertura incluye dos de los elementos sugeridos | La apertura incluye uno de los elementos sugeridos | NP |
| El contenido de la presentación evidencia conocimiento del proceso llevado a cabo durante el desarrollo del proyecto de programación | El contenido de la presentación evidencia conocimiento de las 5 etapas del proceso llevado a cabo durante el desarrollo del proyecto de programación | El contenido de la presentación evidencia conocimiento de dos o tres etapas del desarrollo del proyecto de programación | El contenido de la presentación evidencia conocimiento de 1 de las etapas del desarrollo del proyecto de programación | NP |
| El contenido de la presentación evidencia dominio y comprensión de los conceptos de programación (listas, funciones, condicionales, ciclos) | El contenido de la presentación evidencia dominio y comprensión de los conceptos de programación estudiados | El contenido de la presentación evidencia conocimiento de algunos conceptos de programación | El contenido de la presentación evidencia comprensión de 1 de los conceptos de programación | No evidencia comprensión de los conceptos |
| El material de apoyo complementa el mensaje oral y aporta elementos importantes para su comprensión | Toda la información e imágenes son claras y coherentes con el tema | Dos o tres elementos de la información e imágenes no son claros ni coherentes con el tema | Más de tres elementos de información e imágenes no son claros ni coherentes respecto del tema | NP |
| Evidencian la incorporación de observaciones brindadas por el docente (parte técnica y/o comunicación oral) | Todas las observaciones fueron incorporadas | Una o dos observaciones no fueron incorporadas | Tres o más observaciones no fueron incorporadas. | NP |
| Realizan un manejo adecuado del tiempo | La presentación finaliza en el tiempo establecido. | La presentación supera el tiempo establecido. | La presentación termina antes del tiempo mínimo establecido. | NP |

| | | | | |
|---|---|---|---|----|
| Trabajo en equipo (distribución del tiempo entre los contenidos de la presentación) | La presentación evidencia una distribución equilibrada del tiempo y de los contenidos | En algunos segmentos de la presentación se requiere una mejor distribución del tiempo y de los contenidos | La presentación es poco equilibrada en la distribución del tiempo y de los contenidos | NP |
| Realizan un cierre oportuno y efectivo según los aspectos sugeridos (síntesis de las ideas, reflexión sobre limitaciones, espacio para preguntas y comentarios). | El cierre incluye todos los aspectos sugeridos. | El cierre incluye dos de los aspectos sugeridos. | El cierre incluye uno de los aspectos sugeridos. | NP |
| Observaciones | | | | |

Instrumento de evaluación individual (12 puntos)

| Criterio | 3 | 2 | 1 | 0 |
|---|---|---|--|---|
| Emplea el lenguaje corporal para complementar el mensaje oral (postura erguida, ademanes utilizados acorde con el mensaje, contacto visual durante toda la presentación) | El estudiante aplica correctamente los tres elementos indicados | El estudiante aplica correctamente dos de los elementos indicados. | El estudiante aplica correctamente uno de los elementos indicados. | No presenta o no cumple con el indicador mínimo |
| Habla con claridad, fluidez y volumen | Durante toda la presentación, habla con volumen alto. Se expresa con claridad y fluidez | Durante algunos segmentos de la presentación, habla con volumen alto, se expresa con claridad y fluidez | Durante la mayor parte de la presentación, faltó claridad, fluidez y volumen adecuados. | NP NC |
| Comunica el mensaje desde distintos espacios del escenario (aplica para la segunda y tercera fases) | Durante todo el tiempo de la presentación se aprovecha el espacio para dirigirse al público | Durante algunos segmentos de la presentación se aprovecha el espacio para dirigirse al público | Durante la mayor parte de la presentación se aprovecha el espacio para dirigirse al público | |
| Emplea la cortesía verbal para dirigirse a otros expositores y miembros del público | Durante toda la presentación, el tono de la voz, la expresión del rostro y la selección de las palabras proyectan respeto y cordialidad | Durante algunos segmentos de la presentación, el tono de la voz, la expresión del rostro o las palabras evidencian molestia o irrespeto | Durante la mayor parte de la presentación, el tono de la voz, la expresión del rostro o las palabras evidencian molestia o irrespeto | |
| Observaciones | | | | |

