

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Introducción a la Programación y Computación 2
Escuela de Vacaciones Diciembre 2024



Ing. Jose Manuel Ruiz Juarez
Tutor de curso: Rodrigo Alejandro Hernández de León

PROYECTO 1

OBJETIVO GENERAL

Se busca que el estudiante sea capaz de desarrollar una solución integral que implemente tipos de datos abstractos (TDA) bajo el concepto de programación orientada a objetos.

OBJETIVOS ESPECÍFICOS

- Implementar POO para desarrollo de la solución a través de lenguaje Python
- Utilizar estructuras de programación secuenciales, cíclicas y condicionales.
- Utilizar listas ordenadas para implementar una lista de listas utilizando memoria dinámica
- Visualizar TDA'S por medio de la herramienta Graphviz.
- Utilizar archivos XML como insumos para la lógica y comportamiento de la solución.
- Crear una aplicación de escritorio y llevarla a un sitio web consumiendo una API.

Pasos:

1. Crear el ambiente de desarrollo
2. Feat: carga de archivos artistas > solistas | admin
3. Feat: carga figuras solistas

DESCRIPCIÓN GENERAL

IPCArt-Studio es una plataforma innovadora dedicada a la creación y gestión de arte en píxeles, diseñada para satisfacer las necesidades de artistas digitales y entusiastas del pixel art. Su objetivo principal es proporcionar a los usuarios una herramienta integral que facilite la creación, edición y gestión de obras de pixel art, al tiempo que fomenta la colaboración y el intercambio creativo.

La plataforma ofrece una variedad de funcionalidades que permiten a los usuarios gestionar sus proyectos de pixel art de manera eficiente. Los usuarios pueden crear y editar imágenes utilizando una interfaz intuitiva que soporta tanto el modo de color normal como la escala de grises, permitiendo alternar entre estas versiones de manera fluida. Las imágenes se almacenan de manera optimizada utilizando matrices dispersas, lo que garantiza un uso eficiente de los recursos y una manipulación ágil de grandes obras.

IPCArt-Studio también gestiona una galería de imágenes donde los usuarios pueden navegar a través de sus proyectos completados. Esta galería está implementada como una lista doble circular, lo que permite una navegación continua y sin interrupciones, facilitando la revisión y edición de obras anteriores.

Para facilitar la gestión de solicitudes de pixel art, la plataforma utiliza una pila para acumular imágenes que los usuarios desean solicitar. Esta funcionalidad permite a los usuarios revisar y modificar su selección antes de enviarlas a la cola de solicitudes, donde se procesan en orden de llegada. La cola de solicitudes asegura que cada petición sea atendida de manera justa y eficiente, optimizando el flujo de trabajo.

Además, IPCArt-Studio cuenta con una lista de empleados gestionada mediante una lista simple enlazada, lo que permite una fácil administración de los artistas disponibles para trabajar en los proyectos. Los usuarios, por su parte, se gestionan a través de una lista doble enlazada, facilitando el seguimiento de sus datos de contacto y el historial de solicitudes.

La plataforma no solo se centra en la creación de arte, sino que también promueve la colaboración y el aprendizaje. Ofrece recursos educativos y oportunidades para que los usuarios compartan sus conocimientos y habilidades, contribuyendo al crecimiento de la comunidad de pixel art.

Para llevar una gestión eficiente de todas las actividades que IPCArt-Studio realiza, se requiere la implementación de una aplicación robusta. Usted, como estudiante de alto rendimiento, ha sido contratado para desarrollar esta aplicación, asegurando que todas las funcionalidades mencionadas se integren de manera efectiva para ofrecer una experiencia de usuario excepcional.

IMPLEMENTACIÓN

Como primera fase se requiere de una aplicación de escritorio que tenga las siguientes funcionalidades:

LOGIN

En este apartado únicamente inician sesión los usuarios que solicitaran su diseño de Pixel Art, los artistas o el administrador y tiene que ser la primera vista en la cual el usuario mire cuando abra la aplicación.



Fuente: Elaboración propia

Para ingresar con IPCArt-Studio se tomará en cuenta las siguientes consideraciones:

- El administrador iniciará sesión con el usuario "AdminIPC" y la contraseña "ARTIPC2"
- Los usuarios de tipo Artista comenzaran con las palabras de ART-#ID.
- Los usuarios de tipo Solicitante comenzaran con las palabras de IPC-#ID.

Consideraciones :

- No habrá registro.
- Los usuarios (artistas y solicitantes) se registrarán por medio de la carga en el módulo administración.
- Se seguirá una convención para los nombres de usuario.

MODULO ADMINISTRADOR

En este módulo únicamente ingresa el administrador y contará con las siguientes funcionalidades:

CARGA MASIVA

La aplicación contará con una opción que permitirá que el administrador pueda cargar archivos de tipo XML donde se pueden cargar de 1 a muchos archivos XML y no borrarán el contenido que tengan en las listas al cargar un nuevo archivo XML. Dentro de los archivos XML que se cargarán serán los siguientes:

- Solicitantes:

```
<?xml version="1.0" encoding="UTF-8"?>
<solicitantes>
  <solicitante id="IPC-001" pwd="1234">
    <NombreCompleto>Juan Pérez</NombreCompleto>
    <CorreoElectronico>juan.perez@example.com</CorreoElectronico>
    <NumeroTelefono>12345678</NumeroTelefono>
    <Direccion>Ciudad de Guatemala, Guatemala</Direccion>
  </solicitante>
  ...
</solicitantes>
```

Los usuarios se guardarán en una **Lista Doblemente Enlazada** y cada usuario contará con los datos de:

- Id: Comienza con las letras de IPC- y posteriormente con números y va a ser único en toda la aplicación y valide que no se repita el mismo id en un usuario ya que con este id ingresa a la aplicación al iniciar sesión.
- Pwd: Es la contraseña con la que el usuario podrá iniciar sesión.
- Nombre Completo: Nombre del solicitante.
- Correo Electrónico: Es el correo del solicitante y tiene que validar con una expresión regular que si sea la sintaxis correcta del correo electrónico.
- Número de Teléfono: Es el número del solicitante, tiene que validar que sean enteros y que obligatoriamente debe de contener 8 dígitos.
- Dirección: Es la dirección donde vive el solicitante.

- Artistas:

```
<Artistas>
  <Artista id="ART-001" pwd="1234">
    <NombreCompleto>María López</NombreCompleto>
    <CorreoElectronico>maria.lopez@example.com</CorreoElectronico>
    <NumeroTelefono>87654321</NumeroTelefono>
    <Especialidades>Pixel Art, Animación</Especialidades>
    <NotasAdicionales>Disponible para proyectos urgentes.</NotasAdicionales>
  </Artista>
  <!-- Más artistas aquí -->
</Artistas>
```

Los artistas se guardarán en una **Lista Simplemente Enlazada** y cada artista contará con los datos de:

- Id: Comienza con las letras de ART- y posteriormente con números y va a ser único en toda la aplicación y valide que no se repita el mismo id en un usuario ya que con este id ingresa a la aplicación al iniciar sesión.

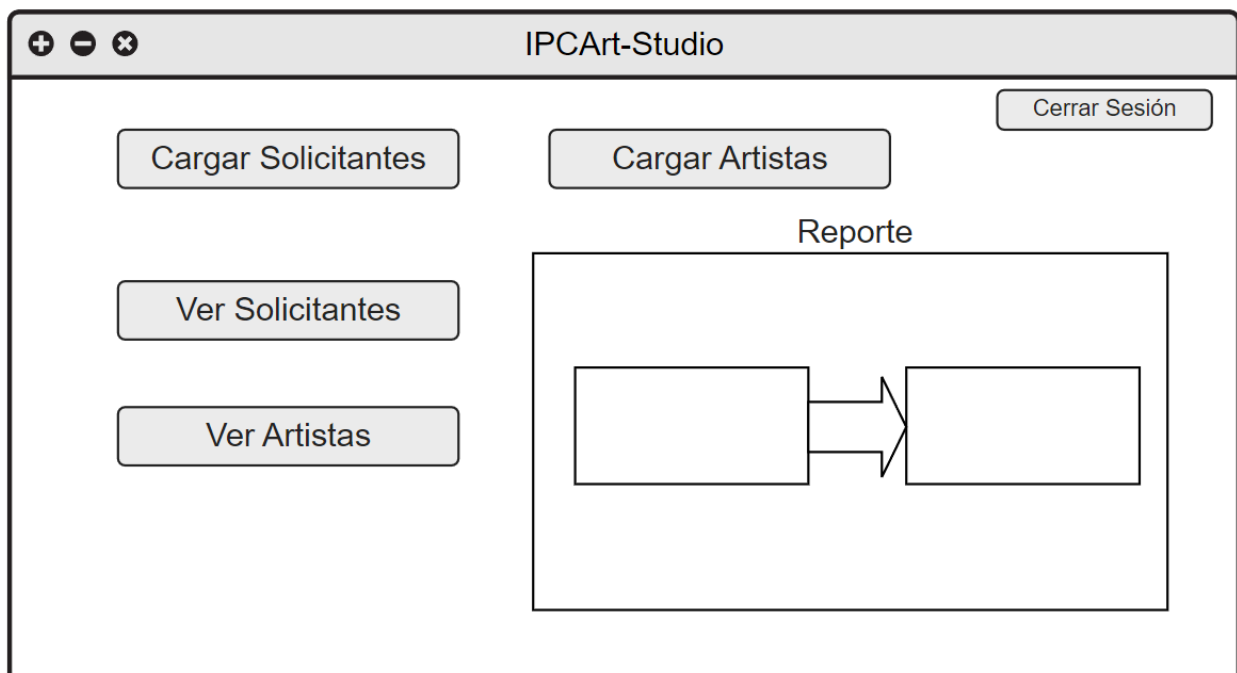
- Pwd: Es la contraseña con la que el artista podrá iniciar sesión.
- Nombre Completo: Nombre del artista.
- Correo Electrónico: Es el correo del artista y tiene que validar con una expresión regular que si sea la sintaxis correcta del correo electrónico.
- Número de Teléfono: Es el número del artista, tiene que validar que sean enteros y que obligatoriamente debe de contener 8 dígitos.
- Especialidades: Describe lo que se especializa el artista.
- Notas Adicionales: Notas adicionales a considerar del artista.

REPORTES

Para verificar que si esté usando TDA'S entonces habrá un apartado de reportes donde en su computadora habrá una carpeta llamada Reportes y ahí se almacenarán todos los reportes solicitados (para los reportes se le solicita que esté graficado con Graphviz):

- **Lista Doblemente Enlazada:** Se abrirá la imagen de la lista doblemente enlazada de usuarios teniendo como ruta `./Reportes/ListaSolicitantes.svg` (la extensión de la imagen lo ideal es que sea .png, .jpg o .svg) y cada nodo debe contener los atributos de cada solicitante.
- **Lista Simplemente Enlazada:** Se abrirá la imagen de la lista simplemente enlazada de artistas teniendo como ruta `./Reportes/ListaArtistas.svg` (la extensión de la imagen lo ideal es que sea .png, .jpg o .svg) y cada nodo debe contener los atributos de cada artista.

Estos se visualizarán en la interfaz de Tkinter, una sugerencia de la interfaz de administrador es la siguiente.



Fuente: Elaboración Propia

MODULO DE ARTISTAS

En este módulo únicamente ingresan los artistas que iniciaron sesión en la aplicación y contendrá lo siguiente:

SOLICITUDES

Los artistas pueden ver la cola de solicitudes por parte de los solicitantes y puede aceptar la solicitud de dibujo de PixelArt y guardar la solicitud en una lista Circular.

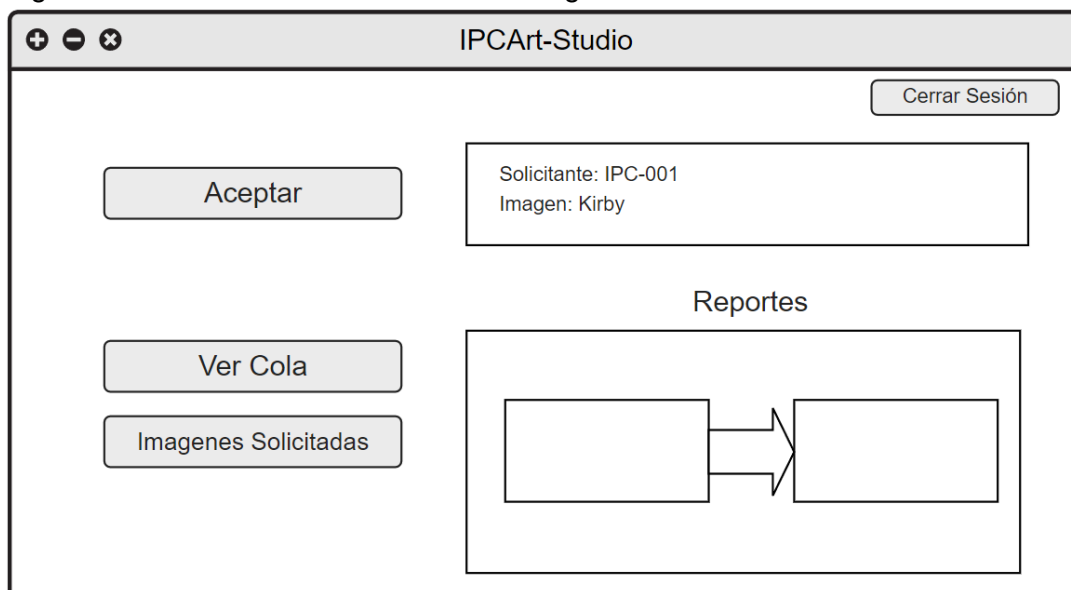
Al darle aceptar, también procederá a graficar la matriz dispersa de la imagen solicitada por el usuario y se almacenará en la lista doblemente circular del usuario. Y se guardará con la ruta de *./Reportes/Matriz_IDFigura.svg* (la extensión de la imagen lo ideal es que sea .svg, .png o .jpg)

REPORTE

Los artistas pueden presionar el botón de VerCola donde se abrirá una imagen la cuál su ruta será *./Reportes/Cola.svg* (la extensión de la imagen lo ideal es que sea .svg, .png o .jpg) y cada nodo debe contener los atributos del nodo de la cola.

Los artistas pueden presionar el botón de “Imágenes Procesadas” donde se abrirá una imagen que contiene la lista circular la cuál su ruta será *./Reportes/Procesadas_IDArtista.svg* (la extensión de la imagen lo ideal es que sea .svg, .png o .jpg) y cada nodo debe contener los atributos del nodo de la lista.

Una sugerencia de la interfaz de usuario es la siguiente:



Fuente: Elaboración Propia

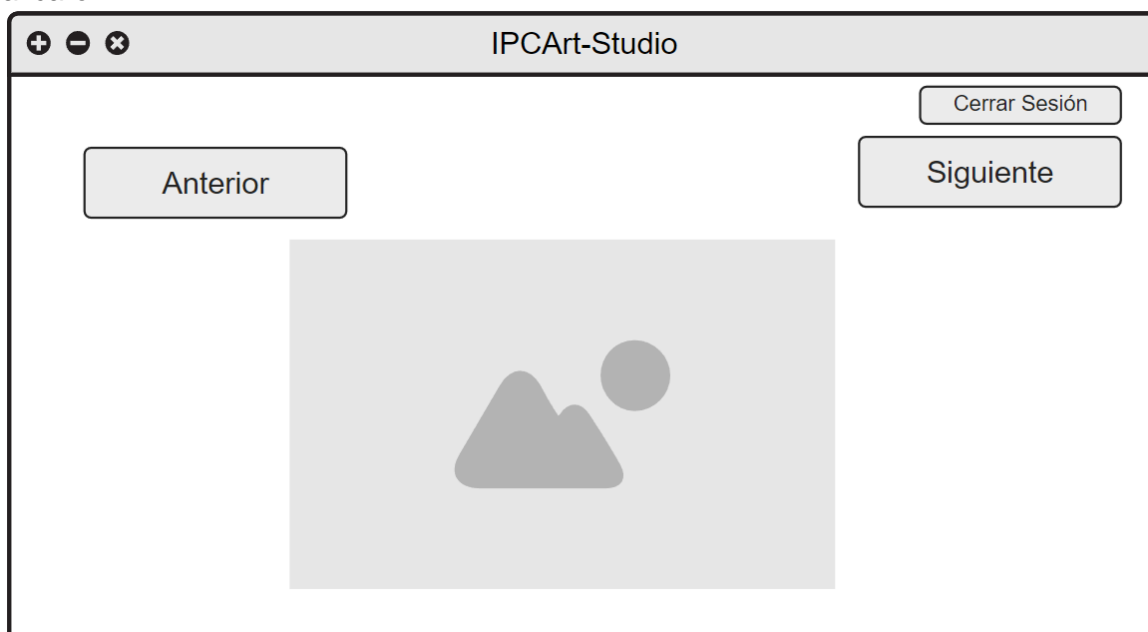
MODULO DE SOLICITANTE

En este módulo únicamente ingresan los solicitantes que iniciaron sesión en la aplicación y contendrá lo siguiente:

GALERIA

Los solicitantes pueden ver una galería de imágenes en la cual se encuentran en una lista circular doblemente enlazada. El solicitante puede moverse hacia adelante o hacia atrás conforme a las posiciones de las imágenes que se encuentre en la lista. Únicamente podrá visualizar las imágenes que han sido solicitadas por ese solicitante y sean generados por el artista.

Las imágenes que se irán mostrando serán las matrices dispersas de las imágenes que se graficaron.



Fuente: Elaboración Propia

SOLICITAR

Los solicitantes pueden solicitar a los artistas que creen sus imágenes en base a una carga de XML. El XML que tiene que cargar el solicitante tiene que contener la siguiente estructura:

```
<?xml version="1.0" encoding="UTF-8"?>
<figura>
  <nombre id="1">Figura1</nombre>
  <diseño>
    <pixel fila="0" col="2">#000000</pixel>
    <pixel fila="0" col="3">#000000</pixel>
    <pixel fila="0" col="4">#000000</pixel>
    <pixel fila="1" col="1">#FFFFFF</pixel>
    <!-- Más pixeles aquí -->
  </diseño>
</figura>
```

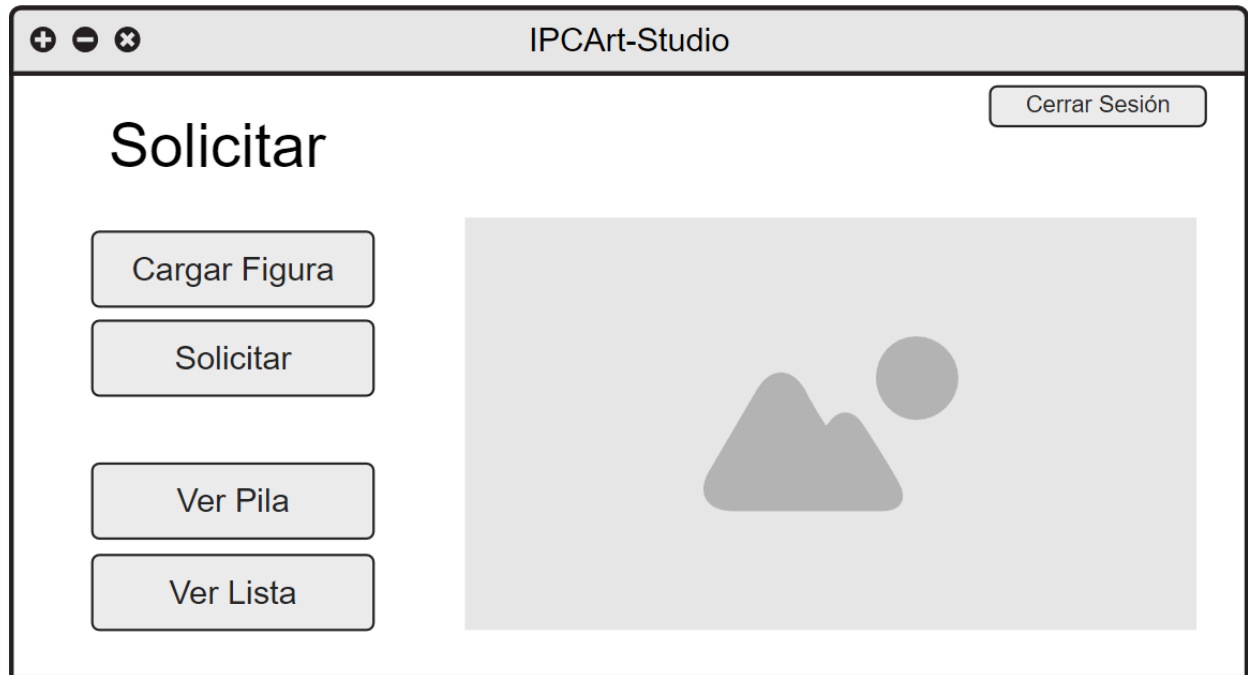
Al cargar la figura, puede cargar otra e irla acumulándolo todo dentro de una pila de tipo carrito de figuras. Hasta que el solicitante se sienta listo, procede a enviar la solicitud a los artistas que se encuentran en el sistema.

REPORTE

Los solicitantes pueden presionar el botón de VerPila donde se abrirá una imagen la cuál su ruta será `./Reportes/Pila_IDSolicitante.svg` (la extensión de la imagen lo ideal es que sea .svg, .png o .jpg) y cada nodo debe contener los atributos del nodo de la pila.

Los solicitantes pueden presionar el botón de “Ver Lista” donde se abrirá una imagen la cuál su ruta será `./Reportes/Lista_Doble_IDSolicitante.svg` (la extensión de la imagen lo ideal es que sea .svg, .png o .jpg) y cada nodo debe contener los atributos del nodo de la lista.

Una sugerencia de la interfaz de usuario es la siguiente:



Fuente: Elaboración Propia

Una sugerencia de la figura graficada con matriz dispersa es la siguiente:



Fuente: Elaboración Propia

CONSIDERACIONES

Debe utilizarse versionamiento para el desarrollo del proyecto. Se utilizará la plataforma Github en la cual se debe crear un repositorio en el que se gestionará el proyecto. Se debe realizar dos releases: el primero realizado el 12 de diciembre con nombre (v.0.1.0) y el segundo realizado antes de la fecha de entrega con nombre (v1.0.0). Además, cada estudiante debe trabajar una rama la cual tendrá de nombre *develop*(*no_carnet*). A la hora de entregar deben de unir sus cambios de la rama *develop* a la rama *main* y con ello poder generar su release. También debe de aplicar las buenas prácticas para realizar commits aprendidas en laboratorio. Para la realización de la interfaz gráfica queda a discreción del estudiante que librería utilizar.

DOCUMENTACIÓN

Para que el proyecto sea calificado, el estudiante deberá entregar la documentación utilizando el formato de ensayo definido para el curso. En el caso del proyecto, el ensayo puede tener un mínimo de 4 y un máximo de 7 páginas de contenido, este máximo no incluye los apéndices o anexos donde se pueden mostrar modelos y diseños utilizados para construir la solución. Este informe debe expresar con claridad el diseño de objetos ideado para resolver este proyecto por lo que debe expresar el diagrama de clases y los diagramas de flujo de los procesos más importantes. Debe de tomar en cuenta que es un ensayo formal, por lo que se calificará tanto la redacción, como ortografía y presentación. Al no presentar el formato del ensayo del curso o no presenta la documentación, no tendrá derecho a nota del proyecto.

RESTRICCIONES

- Solo se permitirá la utilización de los IDEs discutidos en el laboratorio.
- Las listas deben ser creadas completamente por el estudiante mediante clases. Caso contrario, se penalizará con el 100% de la nota.
- Uso obligatorio de programación orientada a objetos (POO).
- El nombre del repositorio debe de ser IPC2_ProyectoVD2024_#carnet. Trabajando dentro de la carpeta llamada "Proyecto 1".
- Agregar al auxiliar al repositorio: **rodrialeipc**. Si al día de la entrega, no se encuentra el auxiliar agregado al laboratorio, no tendrá derecho a calificación.
- Se trabajará de forma individual.
- El estudiante debe subir la documentación solicitada al repositorio para poder optar a la calificación, si entrega otro formato que no sea el ensayo del curso no se calificará el proyecto.
- Los archivos de entrada no podrán modificarse.
- Se calificará en base al release realizado previo a la fecha de entrega. No se calificará dado que se dé el caso que existan modificaciones de código en fechas posteriores a la entrega.
- Se penalizará con el 50% de la nota por cada release no realizado.
- Para dudas concernientes al proyecto se utilizarán los foros en UEDI de manera que todos los estudiantes puedan ver las preguntas y las posteriores respuestas.
- De no existir una forma gráfica de poder validar el funcionamiento de la aplicación se penalizará con el 100% de la nota.
- **COPIAS TOTALES O PARCIALES SERÁN REPORTADOS A LA ESCUELA Y OBTENDRÁN NOTA DE 0 PUNTOS.**
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el domingo 17 de diciembre antes de las 23:59.
- La entrega será por medio de la UEDI.
- Se debe de entregar el link del repositorio.