Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Estructuras de Datos
Primer Semestre de 2023
Proyecto Fase 3

Ingenieros:
- Ing. Edgar Ornelis
- Ing. Álvaro Hernández
- Ing. Luis Espino

Auxiliares:
- Aldo Perez
- Leonardo Martínez
- Cristian Suy

EDD GoDrive

FASE III

Ciudad de Guatemala, 10 de Abril de 2023

Objetivos

Objetivo general:

- Aplicar los conocimientos del curso Estructuras de Datos en el desarrollo de las diferentes estructuras de datos y los diferentes algoritmos de manipulación de la información en ellas.

Objetivos específicos:

- Utilizar el lenguaje JavaScript para implementar estructuras de datos complejos y temas de seguridad, y HTML como interfaz gráfica.
- Utilizar la herramienta Graphviz para graficar las estructuras de datos y html para visualizar imágenes.
- Definir e implementar algoritmos de inserción, encriptación de datos y seguridad

Resumen de estructuras a utilizar:

- Grafo Dirigido: Manejo de carpetas
- Tabla Hash: Almacenamiento de Usuarios
- BlockChain: seguridad y encriptación de envío de mensajes y manejo de datos

Definición del problema

La facultad de ingeniería desea tener una forma de almacenar archivos importantes, pero actualmente no cuentan con un sistema que se ajuste a sus necesidades por lo que se plantea la solución de crear un sistema propio. Este sistema debe de ser capaz de ser utilizado en cualquier sistema operativo por lo que se plantea la posibilidad de que la aplicación sea un sitio web mediante el uso de Github Pages. El sistema debe tener un funcionamiento similar a Google Drive con la característica que la Universidad de San Carlos sea propietario del mismo, además de otras características propias de la aplicación.

Según los requerimientos antes mencionados se desea que usted como estudiante de ingeniería en sistemas desarrolle la aplicación para el manejo de archivos para la Universidad de San Carlos de Guatemala de la facultad de Ingeniería. El sistema denominado como EDD GoDrive debe de llevar el control de usuarios, donde cada uno de los cursos de la carrera de ingeniería en sistemas debe de contar con un espacio de almacenamiento donde se puede subir, crear y eliminar carpetas, así como también archivos. La aplicación debe de ser responsiva y amigable al usuario. A continuación, se da una explicación más detallada de lo solicitado.

Descripción de la aplicación

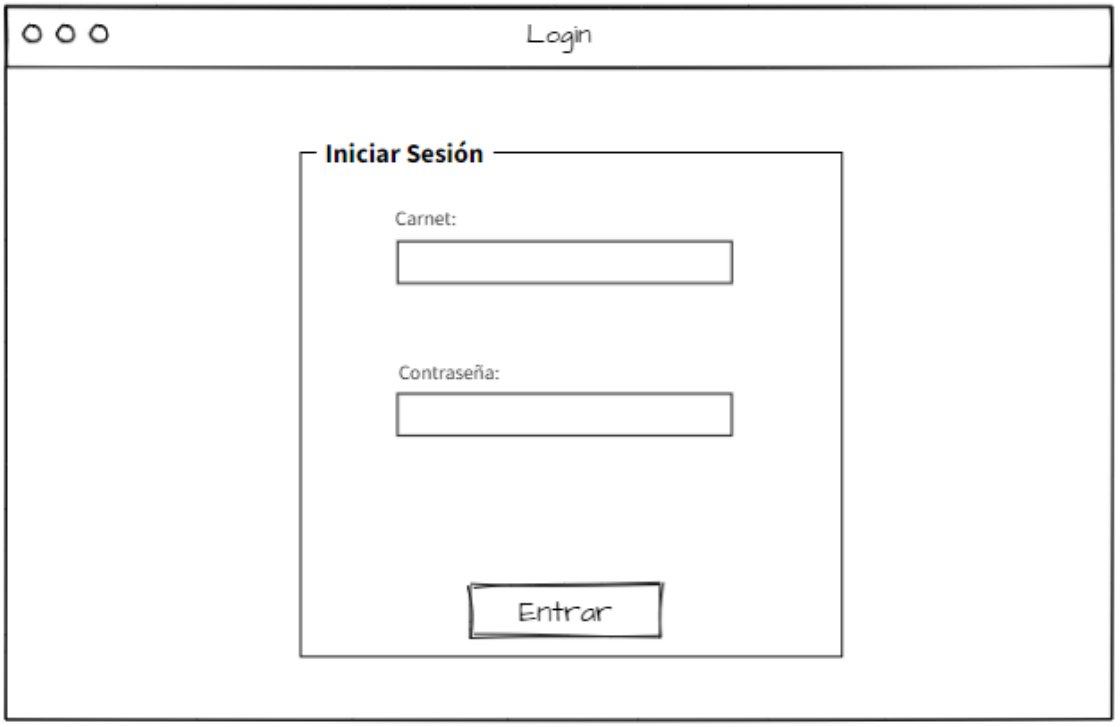
Encriptación

Es una forma de codificar los datos para que solo las partes autorizadas puedan entender la información por lo que se solicita al estudiante que las contraseñas de los usuarios registrados y los mensajes entre usuarios sean almacenados encriptados. Al momento de realizar cualquier acción o mostrar la información al usuario estos se deben desencriptar. Para esto se permite el uso de cualquier librerías o bibliotecas de encriptación, como por ejemplo crypto de javascript, u otra librería que el alumno desee.

Para encriptar/desencriptar queda a discreción del estudiante el método con el cuál será realizado. Tomar en consideración que si utilizan localstorage este tiene un límite por lo que usar un método de encriptación que genere demasiado texto llenará rápido su almacenamiento.

Página de Inicio

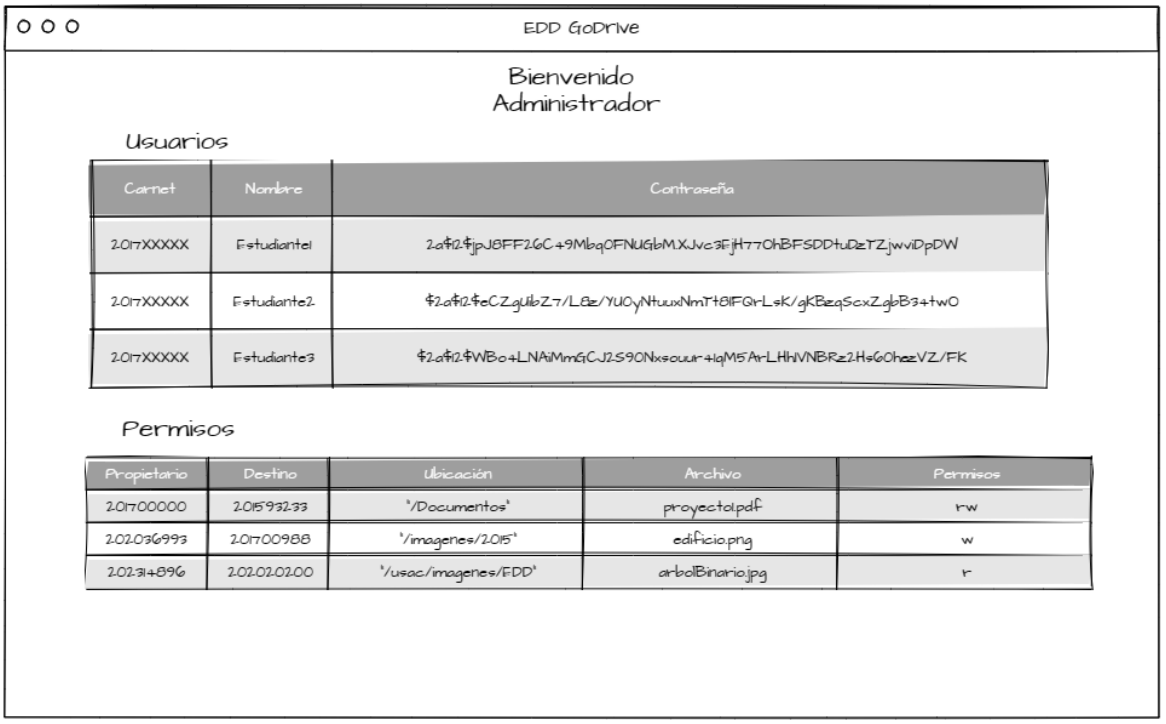
La aplicación contará con una página principal con el cual los estudiantes aceptados podrán iniciar sesión, así mismo el administrador, **teniendo los datos ya almacenados en un árbol AVL (fase 2)**, ahora todos los alumnos se almacenarán en una **tabla hash** bajo las siguientes condiciones, la tabla hash de principio debe tener 7 espacios disponibles, **cuando la tabla llega a un 75% de utilización la tabla aumentará su capacidad hasta el siguiente número primo.** El valor de referencia para insertar en la tabla hash será el carnet del usuario el cual se deberá codificar para obtener un valor entero el cual servirá como parámetro para una función hash por división y así encontrar el índice en el cual se insertará el nodo, **Para las colisiones dentro de la tabla se utilizará una resolución direccionamiento abierto por salto al cuadrado** el cual tomará el valor hash calculado y lo elevará al cuadrado e intentará insertar el nodo en la nueva posición en caso el nuevo hash sobrepase el arreglo se debe de empezar a recorrer el arreglo desde el inicio con los saltos restantes hasta que se encuentre uno vacío.



Funciones de Administrador

Vista Principal

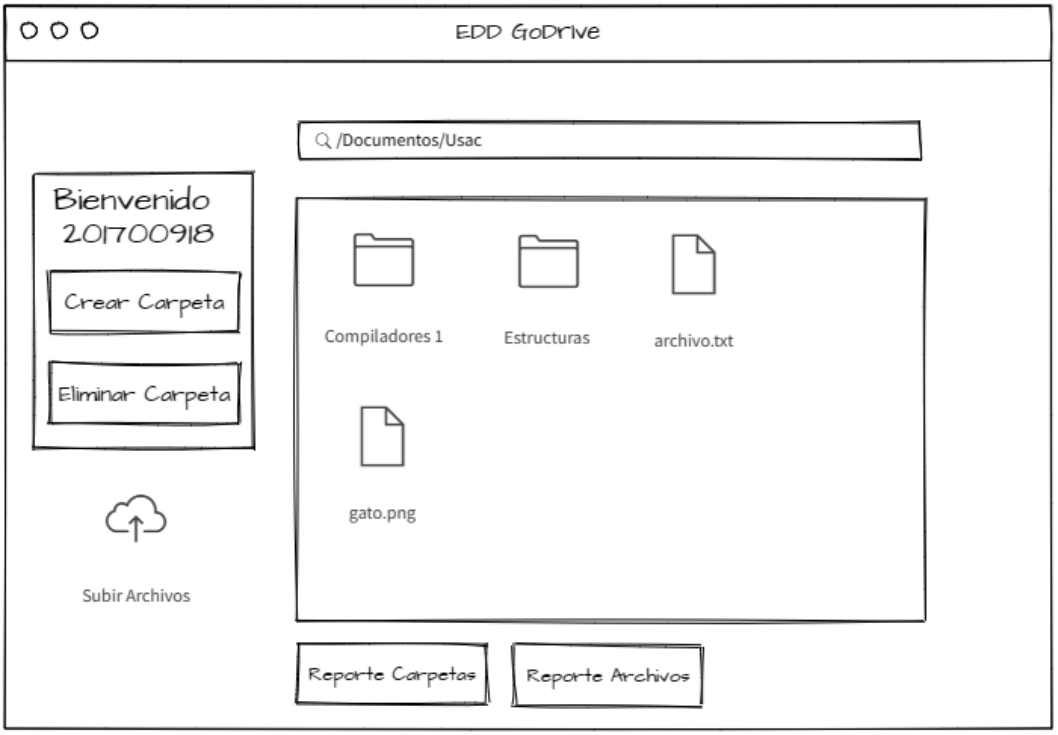
Para las funciones de administrador, este contará con un apartado donde pueda ver visualmente los alumnos que hay dentro del sistema ya aprobados en una tabla, en este caso el administrador podrá visualizar carnet, nombre, y contraseña encriptada para seguridad de los datos, como si de una base de datos se tratara y una tabla general de los permisos de los usuarios, donde se podrá visualizar los archivos compartidos dentro del sistema, este mostrará el Usuario Propietario, el Usuario Destino, Ruta del archivo, Nombre de Archivo, Permiso del archivo. Además tendrá un botón para generar el reporte de mensajes el cual se explicará más adelante.



Funciones de Usuario

Vista Principal

El usuario que inicie sesión tendrá una ventana donde mostrará diferentes opciones de creación, eliminación, modificación de carpetas. Para que el usuario pueda acceder a la carpetas de sus sistema se usará una barra superior de búsqueda, donde colocará la ruta de su carpeta, se debe validar que la ruta de la carpeta sera existente, ejemplo si queremos acceder a **/imagenes/2023** pero la carpeta **imagenes** no existe en raíz, debe mostrar un alerta que especifique que el directorio no es válido. **Para el almacenamiento de las carpetas se usará el árbol indexado de la fase 2 para leer las carpetas del sistema y se almacenará en una matriz de adyacencia para poder realizar el árbol de recubrimiento, para visualizar las relaciones entre directorios, el cual servirá para la navegación entre carpetas, tomando como carpeta raíz "/" el nodo principal del árbol.**



<u>Pestaña de compartidos</u>

El usuario tendrá un apartado nuevo llamado Compartidos Conmigo en el cual podrá visualizar los archivos compartidos, donde se mostrará el contenido del mismo, en el caso de los PDF se usará un **Iframe**, en el caso de imágenes será **img** utilizando la data cargada de la fase 2, y por último en el caso de archivos de texto será en un **textarea**

<u>Sistema de Mensajería</u>

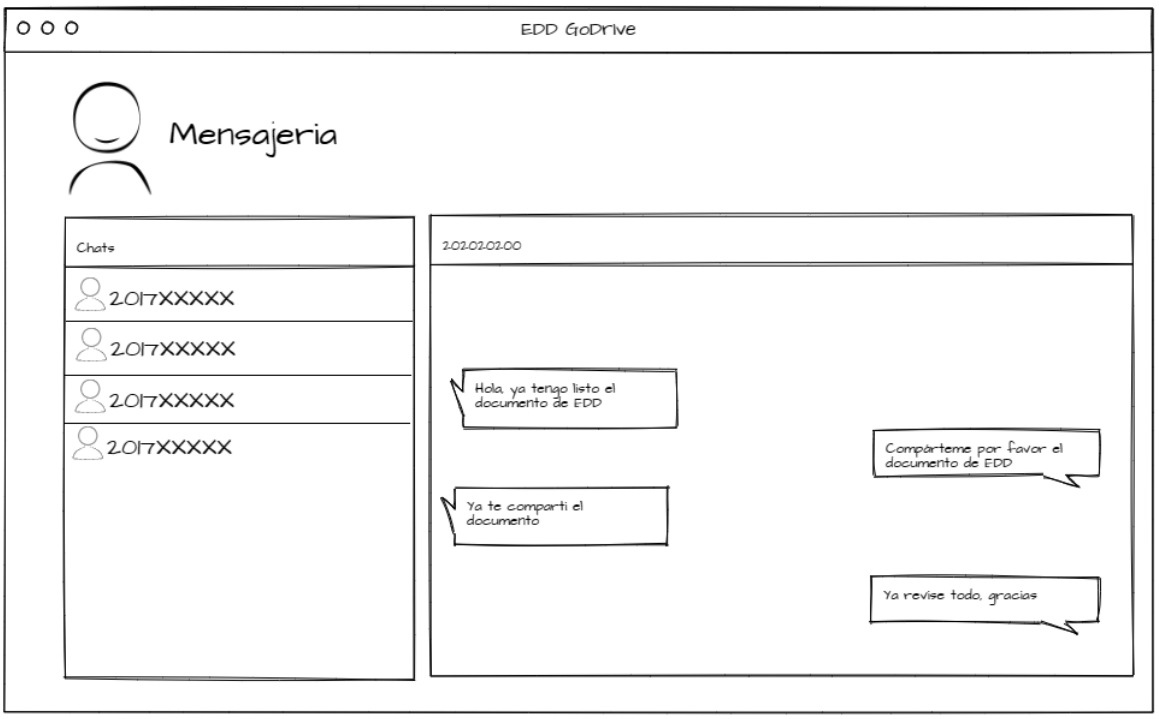
Con el fin de ayudar la comunicación entre los estudiantes, se realizará un apartado nuevo donde el estudiante podrá enviar mensajes a otros usuarios, los mensajes para seguridad de la conversación entre los estudiantes, los mensajes deben ir encriptados, cuando el usuario emisor envía un mensaje, el texto debe ser encriptado y quedarse así hasta que el usuario receptor inicie sesión y pueda revisar los mensajes nuevos y se desencripta el mensaje para su visualización. **Para el manejo de la mensajería se utilizará el sistema de BlockChain para la seguridad e integridad del mismo, para esto se utilizara el siguiente sistema de almacenamiento, su similitud es al de una lista doble enlazada de nodos, el sistema de blockchain contará con los siguientes atributos:**

Index: Este número representa el número del bloque, para el bloque génesis tendrá el valor de index 0 y cada bloque que se cree será tendrá los valores 1, 2, 3, … etc.

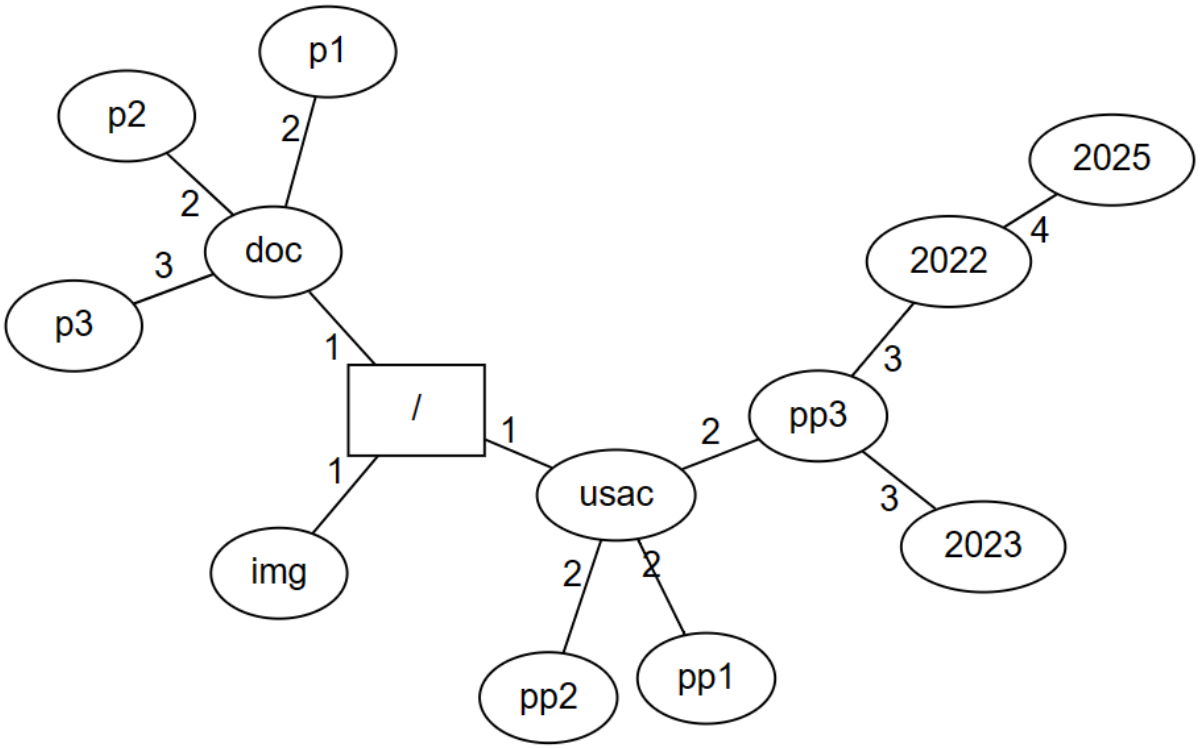
Timestamp: Representa la fecha y hora de creación exacta, este contará con la siguiente estructura (DD-MM-YY-::HH:MM:SS).

Transmitter: Este representará al emisor del mensaje, en este caso el carnet del mismo.

Receiver: Este representará al receptor del mensaje, en este caso el carnet del mismo.
Message: Éste será el cuerpo del mensaje que se envió al usuario, para éste deberá aplicarse la encriptación AES.
PreviousHash: Este es el bloque previo y sirve para validar que la cadena del blockchain no esté corrupta. para el caso del bloque génesis, hash anterior deberá ser 0000
Hash: Este protege la información del mensaje y que este no este corrupto, para el caso de este se usará la encriptación SHA256 y creará una función para recibir com parametros parametros utilizara index, timestamp, transmitter, receiver, message.
Ejemplo: SHA-256(index+timestamp+transmitter+receiver+message)



<u>Reporte Carpetas</u>
El usuario tendrá la opción de crear el reporte del grafo dirigido, para visualizar las relaciones y conexiones que hay entre carpetas

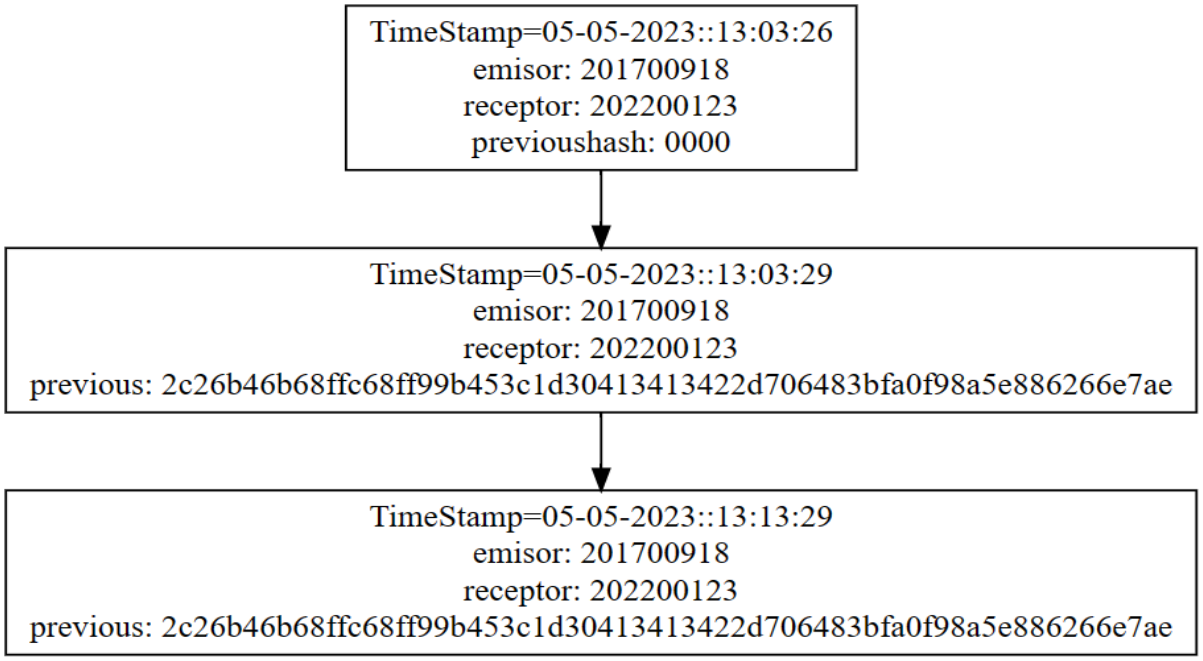


Reporte de Mensajes

El administrador podrá visualizar de manera general todo el sistema de mensajería mediante un reporte, que se muestra a continuación, teniendo en cuenta que el mensaje siempre debe visualizarse encriptado para mayor seguridad, este sera generado un textarea donde cargara esa información con la posibilidad de ver cada bloque, la estructura del texto será la siguiente:

Index: 0 TimeStamp = 05-05-2023::13:03:26 emisor: 201700918 receptor: 202200123 Mensaje: YU7gYqeIBiIVgGemuy/xNnpph/NJi0S8JdSkSIOkjpI= PreviousHash: 0000 Hash: 2c26b46b68ffc68ff99b453c1d30413413422d706483bfa0f98a5e886266e7ae

Y el reporte gráfico, mostrará todo los bloques pero con la siguiente estructura e información



Tecnologías a utilizar

Aplicación web estática:
Se contará con una aplicación web estática alojada en **Github Pages**, desarrollada con **Javascript** y **HTML/CSS**. Ya que los repositorios deben ser privados, es necesario que el estudiante active GitHub Student Developer Pack https://education.github.com/students para usar Github Pages con repositorios privados, utilizar su correo institucional para hacer la solicitud.

Manual para solicitud de GitHub Student Developer Pack (Acceder al documento con correo institucional) Solicitud de Github Student

Graphviz:
Todos los reportes deberán estar realizados con **Graphviz** y se podrán generar en cualquier momento para poder observar el estado actual de las estructuras en la interfaz. Se debe de utilizar https://quickchart.io/documentation/graphviz-api/.

Ejemplo en HTML:

```
<img id="image"
    src="https://quickchart.io/graphviz?graph=digraph{ a -> b }"
/>
```

Restricciones

- Las estructuras deben de ser desarrolladas por los estudiantes **sin el uso de ninguna librería o estructura predefinida en el lenguaje**.
- Los reportes son esenciales para verificar si se trabajaron correctamente las estructuras solicitadas, por lo que si no se tiene el reporte de alguna estructura se anularán los puntos que tengan relación tanto al reporte como a la estructura en cuestión.
- No se permite el uso de ningún framework o librerías cómo Angular o React para el desarrollo del proyecto.
- **Sólo se calificará de lo que esté subido en Github Pages, no se realizará la calificación del sitio en local.**

Observaciones

- El lenguaje para esta fase será **Javascript** y **HTML/CSS** como interfaz gráfica**.**
- Para la secciones de encriptación se permite cualquier librería o biblioteca para su uso
- Para el desarrollo de la interfaz se permite el uso de librerías o frameworks como **Bootstrap o Materialize,** entre otros**.**
- Herramienta de desarrollo de reportes solamente **Graphviz** y **HTML** para visualizar imágenes.
- La aplicación web estática debe estar alojada en Github Pages.
- La entrega se realizará por medio de Github, el nombre del repositorio debe ser **EDD_1S2023_PY_#carnet**, donde se creará una carpeta con el nombre **EDD_Proyecto1_Fase3**. Y por medio de **UEDI** se hará entrega del link de su repositorio.
- Deben de agregar a los auxiliares como colaboradores en los repositorios según su sección:
    - **Sección A: ebany**
    - **Sección B: leonardo0eddb**
    - **Sección C: CristianMejia2198**
- Recordar tener sus repositorios en privado, para evitar copias de código.
- Realizar los **manual de técnico y de usuario** en el README en la carpeta de la respectiva Fase.
- Toda duda que se tenga durante el proceso, será realizada por medio de los foros de **UEDI** de la respectiva sección.
- Fecha de entrega: **5 de Mayo, a las 23:59 horas**.
- Las copias encontradas serán penalizadas con 0 y reportadas a la Escuela de Ciencias y Sistemas.