

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Estructuras de Datos
Primer Semestre de 2023
Proyecto Fase 2



Ingenieros:

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

Auxiliares:

- Aldo Perez
- Leonardo Martínez
- Cristian Suy

EDD GoDrive

FASE II

Ciudad de Guatemala, 26 de febrero 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 lineales y no lineales 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, búsqueda en árboles binarios.

Resumen de estructuras a utilizar:

- Árbol AVL: Usuarios que estén en el sistema ya aceptados.
- Matriz Dispersa: Se utilizará para manejar permisos en las carpetas del directorio actual.
- Árbol Multicamino o N-ario: se usará para el manejo de los sistemas de archivos.
- Lista Circular: Manejo de logs, creación o eliminación de carpetas o archivos ligados a cada usuario

Definición del problema

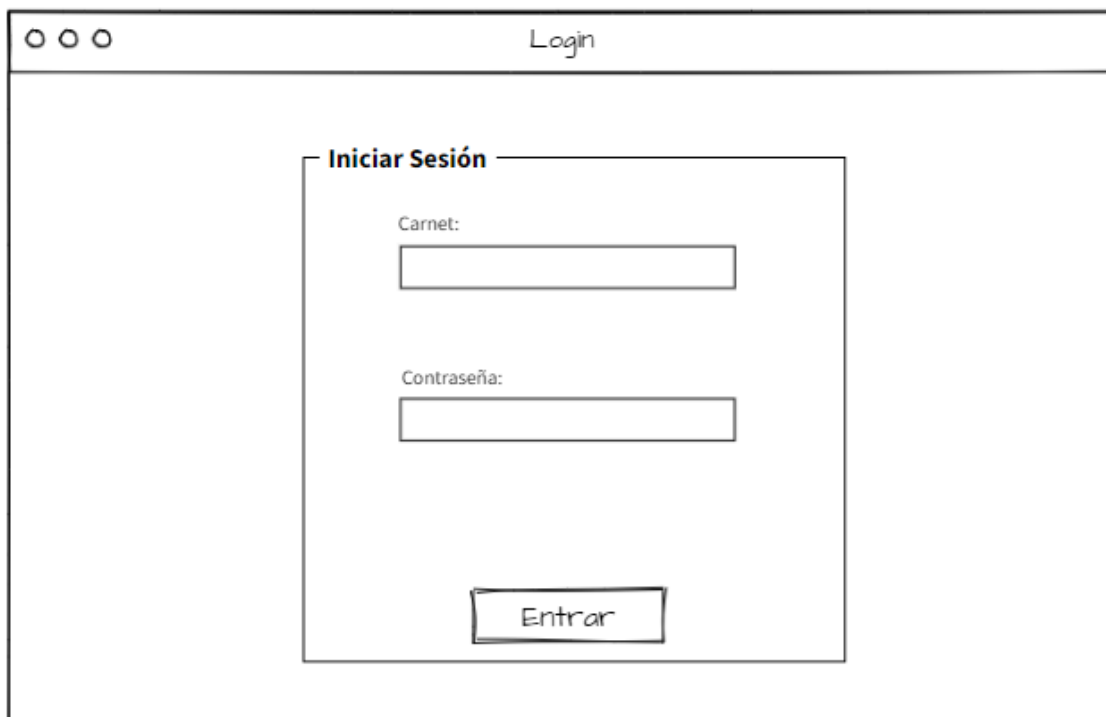
La facultad de ingeniería desea tener una forma de almacenar archivos importantes, pero actualmente no cuentan con un sistema que se ajuste a sus necesidades por lo que se plantea la solución de crear un sistema propio. Este sistema debe de ser capaz de ser utilizado en cualquier sistema operativo por 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. Los usuarios también tendrán la opción de modificar los nombres de carpetas y archivos ya creados. 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

Página de Inicio

La aplicación contará con una página principal con el cual los estudiantes aceptados podrán iniciar sesión, así mismo el administrador, **se deberá implementar la búsqueda en el árbol de los alumnos para encontrar al usuario, en el caso del administrador no estará dentro del árbol, solo se valida si el usuario y contraseña son igual a “Admin”.**

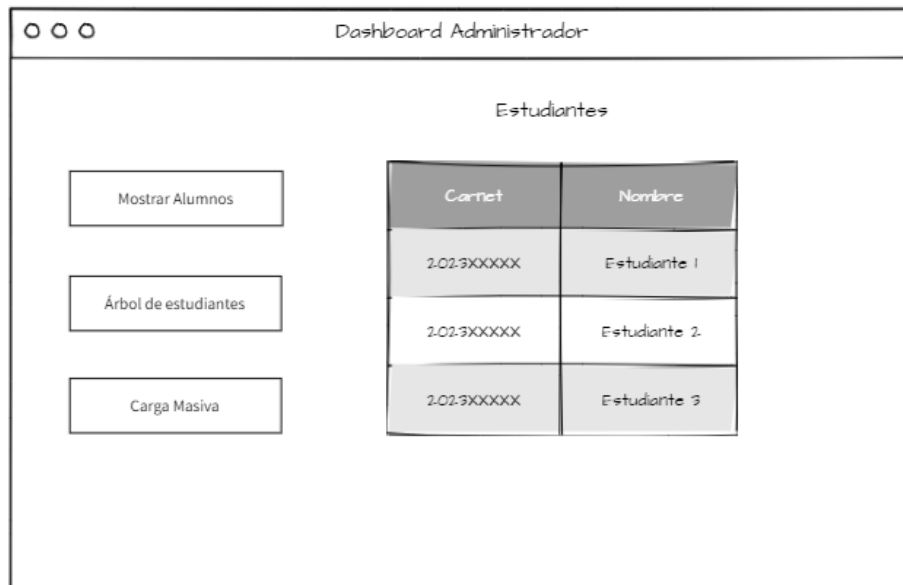


A hand-drawn diagram of a login window. The window has a title bar with three circles on the left and the text "Login" on the right. Inside the window, there is a box titled "Iniciar Sesión". Within this box, there are two input fields: the first is labeled "Carnet:" and the second is labeled "Contraseña:". Below these fields is a button labeled "Entrar".

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, además de esto el administrador podrá generar el reporte del árbol que se genera de los alumnos, **para el almacenamiento de los datos de los alumnos estarán en un árbol AVL que deberán recorrer para mostrar los datos, para esto se usará el recorrido In-Orden por defecto, el administrador podrá elegir si quiere otro tipo de recorrido sea Post-Orden o Pre-Orden.**



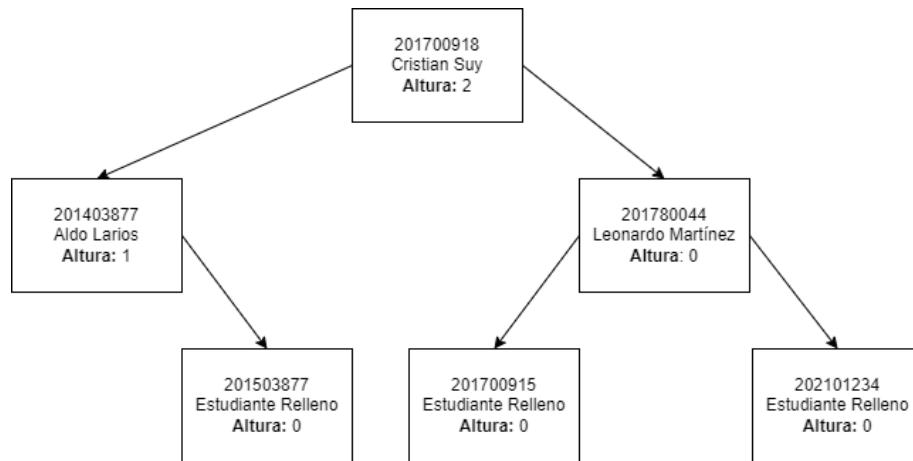
Carga Masiva

El administrador tendrá la opción de subir un archivo de tipo JSON con los datos de los estudiantes, para facilitar el manejo de los estudiantes aceptados.

```
{
  "alumnos": [
    {
      "nombre": "Ebany Larios",
      "carnet": 201403877,
      "password": "qwerty12",
      "Carpeta_Raiz": "/"
    },
    {
      "nombre": "Cristian Suy",
      "carnet": 201700918,
      "password": "cris1234",
      "Carpeta_Raiz": "/"
    },
    {
      "nombre": "Leonardo Martinez",
      "carnet": 201780044,
      "password": "leo1234",
      "Carpeta_Raiz": "/"
    }
  ]
}
```

Reportes

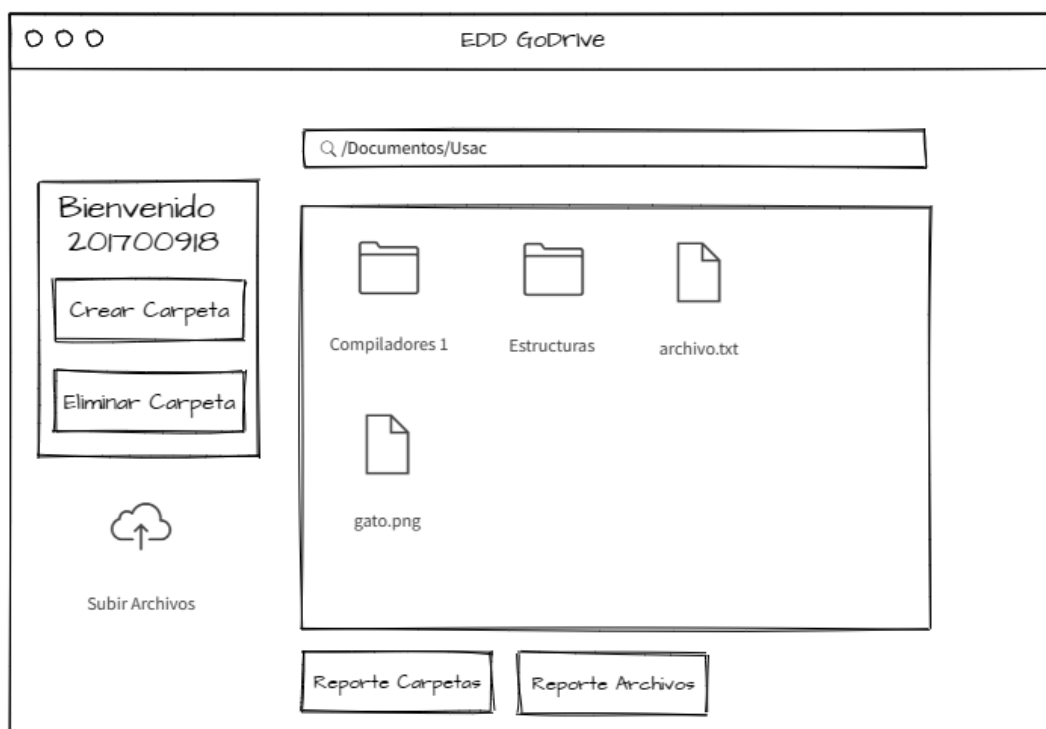
Para las funciones de administrador, este contará con un apartado de reportes en el que se muestre el **Árbol AVL** con todos los estudiantes y la altura que posee cada nodo de la estructura:



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 deberá implementar un árbol indexado, cada usuario dentro del sistema tendrá su propio árbol y para el almacenamiento de los archivos de cada carpeta será una matriz dispersa ligada a cada nodo del arbol.**



Subir Archivo (Sólo .txt, .pdf e imágenes)

En la ventana principal de usuarios tendrá un apartado en el que se pueda subir archivos, se deberá de restringir la extensión de los archivos que se suban a cualquier extensión de imagen, archivos PDF o archivos de texto. Los archivos de imágenes o archivos PDF se tendrán que codificar en **base64** para su almacenamiento en su estructura.

No es posible agregar archivos con el mismo nombre en la misma ruta, si ya existe un archivo con el nombre del archivo a cargar se deberá agregar un alias adicional, por ejemplo: si ya existe “a.txt” y trato de cargar nuevamente “**a.txt**” se le debería asignar “**a(copia1).txt**”, el alias queda a discreción del estudiante.

Crear Carpeta

El usuario tendrá la opción de crear carpeta con el nombre que desee y está a su vez podrá almacenar tanto archivos como otras carpetas, al crear la carpeta tener en cuenta que se realiza en el directorio actual donde esté el usuario, debe validar que la carpeta que se quiere crear no exista ya dentro de la carpeta actual, si existe solo creará una copia, por ejemplo al crear **Estructuras** como ya existe, creará la carpeta con el siguiente nombre “**Copia Estructuras**”

Eliminar Carpeta

El usuario tendrá la opción de eliminar carpeta, se deberá de eliminar tanto la carpeta como todos los archivos y carpetas que posea dentro, para esto la aplicación pedirá el nombre de la carpeta y buscará en la carpeta actual si existe o no la carpeta que se quiera eliminar.

Permisos de archivos

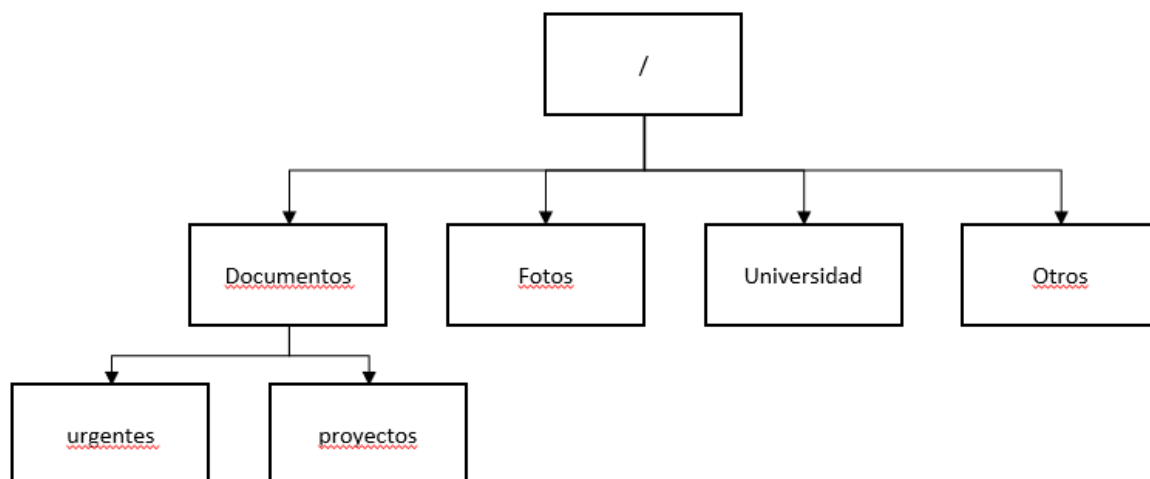
El usuario tendrá la opción de otorgar permisos de sus archivos a otro usuario, para ello se usará una interfaz donde se especifique al usuario, el archivo y los permisos que le dará, para ello se usará los siguientes sufijos:

- r (permisos para leer)
- w (permisos para escribir)

Se usará una matriz dispersa para poder almacenar esta información.

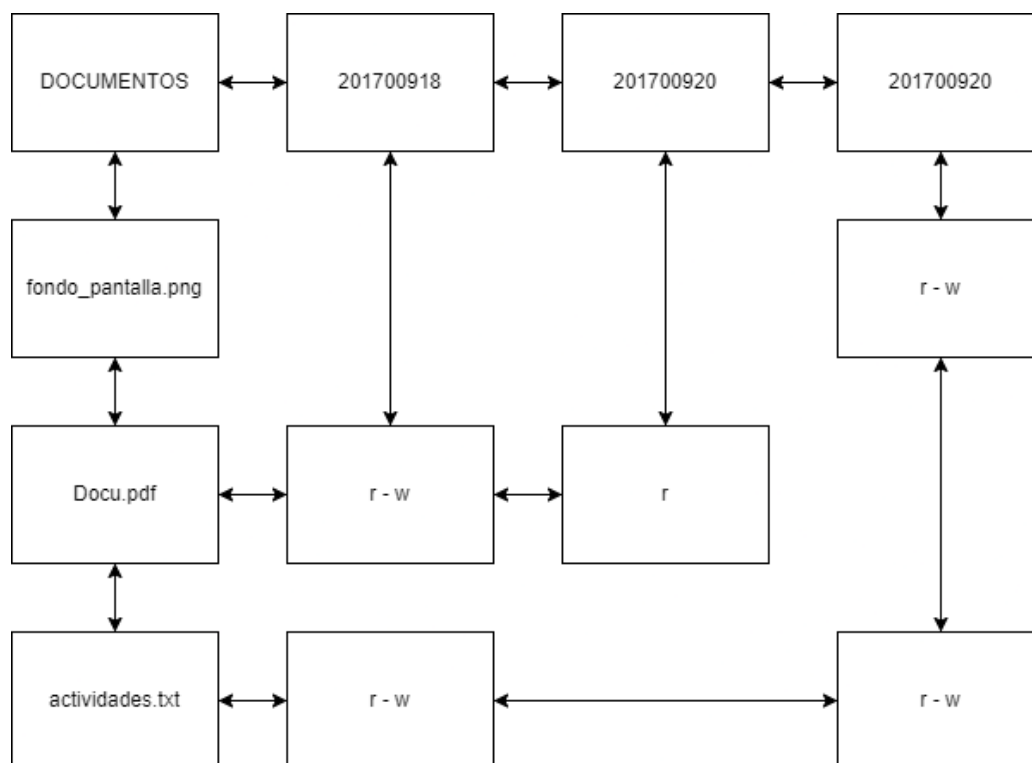
Reporte Carpetas

El usuario tendrá la opción de crear el reporte del árbol N-ario de archivos del usuario logueado.



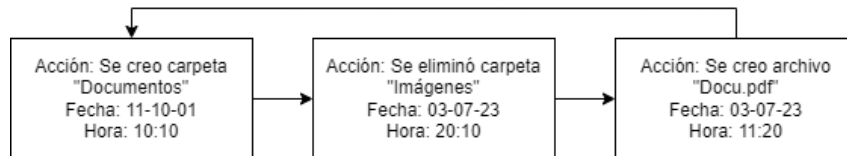
Reporte de Archivos y Permisos

El usuario tendrá la opción de crear el reporte de la matriz que está asociado a los archivos de la **carpeta actual** se debe de mostrar la matriz dispersa con los permisos otorgados por el propietario de los archivos en la carpeta actual que se esté mostrando en la interfaz. Si la carpeta actual no posee ningún archivo, mostrar un mensaje que diga, “no se puede crear la matriz por falta de archivos”.



Reporte de la bitácora:

Está almacenará todas las acciones que realice el usuario que no sea administrador, al momento de crear o eliminar un archivo o carpeta de deberá de mostrar en una **lista enlazada circular**.



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:

```

```

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 como 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 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.
- Deben de tener por lo **menos 2 commits por semana**, de no ser así serán penalizados con -15% de la nota obtenida.
- 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_Fase2**. 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: **4 de Abril, a las 23:59 horas**.
- Las copias encontradas serán penalizadas con 0 y reportadas a la Escuela de Ciencias y Sistemas.