

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



Ingenieros:

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

Auxiliares:

- Aldo Perez
- Leonardo Martínez
- Cristian Suy

EDD ProjectUp

FASE III

Ciudad de Guatemala, 02 de octubre 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 Python para implementar estructuras de datos.
- Utilizar la herramienta Graphviz para graficar las estructuras de datos.
- Definir e implementar algoritmos de búsqueda, relaciones entre estructuras como Grafos, algoritmos en relación a la criptografía y blockchain.
- Desarrollar interfaces gráficas en Python.

Resumen de estructuras a utilizar:

- **Estructuras Fase Anterior:** Se utilizará las estructuras de la fase anterior ya que estas son de almacenamiento principalmente.
- **Grafos:** Relaciones entre tareas.
- **Blockchain (Árbol de Merkle):** Métodos de pagos a empleados utilizado
- **Encriptación:** Para seguridad en contraseñas y Blockchain.

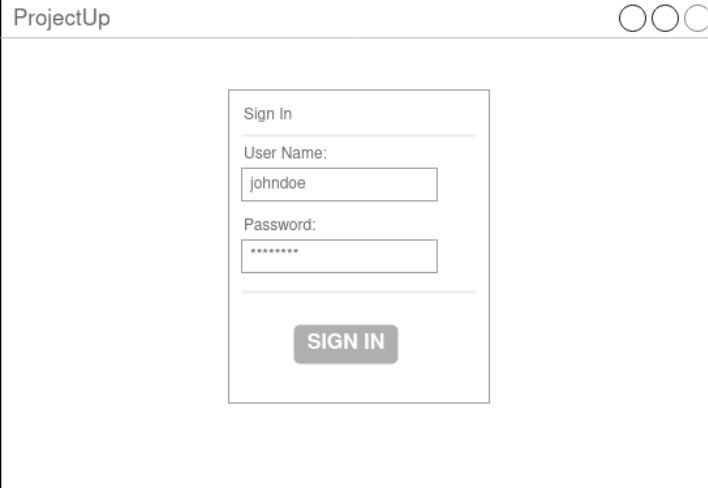
Definición del problema

ProjectUp es una empresa que se dedica al marketing digital, sin embargo vieron el auge tenía el mundo web y cómo eso impulsaba que pequeñas empresa poder promocionar sus pequeños negocios a través de internet además de solo las redes sociales, por lo que la empresa decidió comenzar a realizar páginas web, sin embargo se dieron cuenta que para entregar una página web de calidad, debe tener un equipo completo, por lo que se realizó una investigación para saber que tipo de empleados debe tener el equipo de desarrollo web. Luego de realizar la selección del equipo completo, la empresa ProjectUp se dio cuenta que ellos necesitan un programa que sea capaz de tener control de las tareas para mantener un orden y entregar los proyectos a tiempo sin complicaciones, por lo que solicita que usted como estudiante en sistemas, cree una plataforma en base a sus conocimientos basicos en el ámbito de las metodologías SCRUM, ellos vieron que ya existen plataformas, como Trello para la gestión de proyectos, pero que también ellos como marketing puedan utilizarlo como un Air Table e ir organizando su tareas de marketing, para tener todo conectado y separado segun el area, para ellos se solicita que realice una aplicación de consola dedicado al equipo de desarrollo web para realizar pruebas y ver la eficiencia de la aplicación.

Descripción de la aplicación

Usuario Administrador (Project Manager)

Para iniciar a trabajar en la aplicación, se tendrá ya registrado un project manager dentro del sistema, el cual para ingresar debe tener la estructura de la siguiente manera, PM-#carnet-estudiante, (ejemplo: PM-202300795) y este mismo será usado para el login que se explicará más adelante, es importante resaltar que debe utilizar si o si la estructura del usuario que se especificó. El Project Manager, contará con algunas funciones claves que serán explicadas a continuación:



The image shows a web application window titled "ProjectUp". Inside the window, there is a "Sign In" form. The form has two input fields: "User Name:" with the value "johndoe" and "Password:" with masked characters "*****". Below the password field is a "SIGN IN" button.

Previo a Fase 3:

Para el almacenamiento de la información se usará las estructuras implementadas en la fase anterior, por lo cual se hace resumen de lo necesario para continuar.

- Carga Masiva de empleados, se utilizara el mismo archivo csv de la calificación.
- Carga de archivo JSON, se utilizara el mismo archivo json de la calificación
- El algoritmo de inserción de datos y las estructuras a utilizar son las mismas de la fase anterior.

Administrador:

El administrador realizará toda la carga masiva tal y como en la fase 2, utilizando las mismas estructuras para su almacenamiento. Utilizando los mismos módulos para empleados, proyectos y tareas.

Encriptación de contraseñas:

En este apartado, cuando se realiza la carga de empleados mediante el archivo con extensión .csv se debe realizar una encriptación de contraseñas previo a guardar la información en la tabla hash de la fase anterior, para mantener la integridad y seguridad de los datos al momento de iniciar sesión. El método de encriptación que se utilizara será SHA256, recordando que este método no permite la descriptación de los datos, por lo cual al momento de realizar la comprobación en el inicio de sesión, se debe encriptar la contraseña obtenida en el inicio de sesión y comparar esas cadenas encriptadas.

Pago de empleados:

Cuando los empleados completan las tareas asignadas por cada proyecto, los empleados se les pagará mediante carteras digitales, las cuales están ligadas a cada empleado, para esto se hará la carga de un archivo json, el cual contendrá la el ID del empleado y el código de su Cartera Digital, dicha información será agregada a la tabla hash como parte de un atributo extra.

Los atributos es un arreglo llamado eWallet, y cada posición de ese arreglo tendrá los atributos empleado y id, de la siguiente manera:

```
{
  "eWallet": [
    {
      "empleado": "BDEV-100",
      "id": "DC78ED15"
    },
    {
      "empleado": "FDEV-100",
      "id": "DC38ED15"
    },
    {
      "empleado": "QA-100",
      "id": "JK78ED15"
    }
  ]
}
```

Para realizar el pago, el administrador o Project Manager debe tener una pestaña nueva, donde se muestran los proyectos, y los empleados que están asociados a ese proyecto. Para realizar el pago del mismo y saber cuanto es la cantidad, se tomará de la lista de adyacencia, la cual se explicará más adelante en el apartado de Dependencias de tareas

Guardado de operaciones de pagos:

Para guardar la integridad de los pagos y estos no se vean afectados, se utilizara el árbol de merkle el cual es una estructura fiable para conservar la integridad de datos, el almacenamiento de cada árbol se guardará en cualquier estructura para tener un respaldo de las mismas, queda a discrecion del estudiante que utilizar, para

realizar este árbol se realizara el método SHA-3 para las funciones de hash criptográfico, partirá del siguiente información desde los nodos hojas:

Ejemplo:

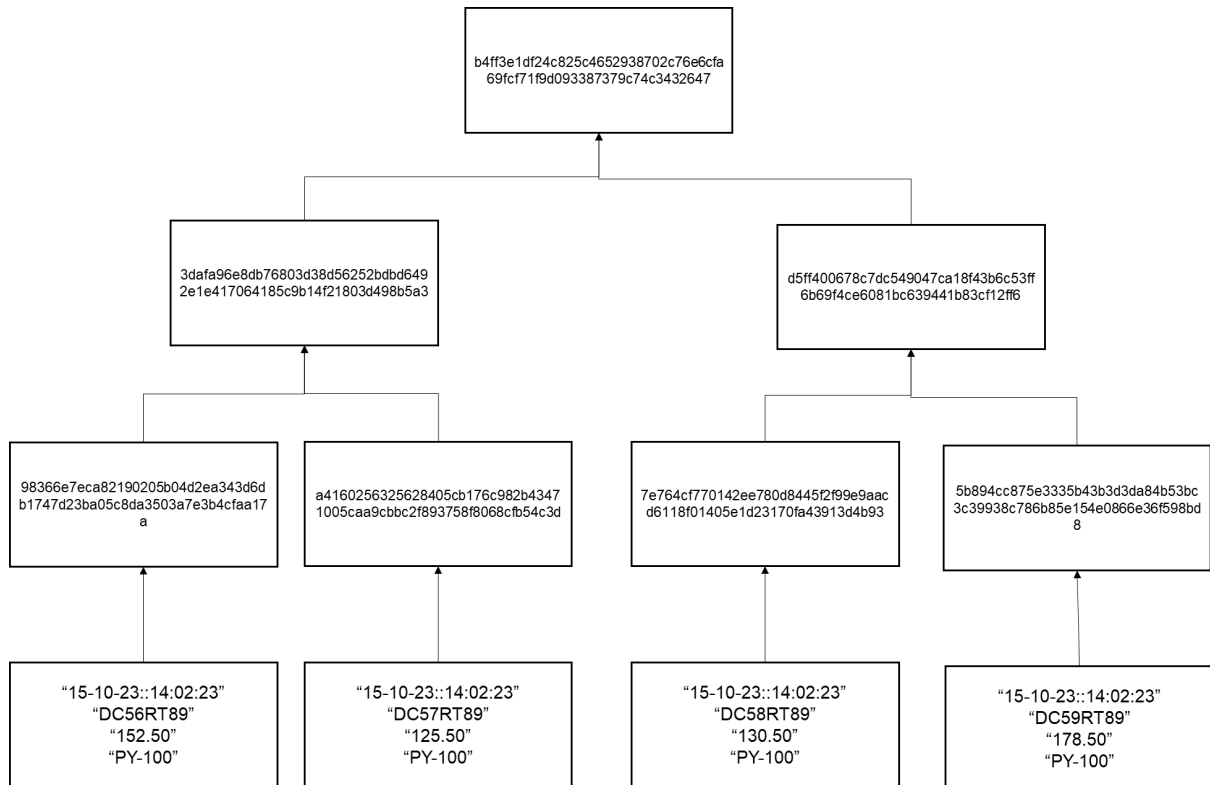
La información que contendrá cada nodo hoja para realizar dicho árbol es la siguiente:

- Fecha y hora de Pago (se tomará la información de la fecha de su computadora, tomando en cuenta el siguiente formato DD-MM-YY::HH:MM:SS)
- ID de la billetera del empleado a cargo
- Monto Pagado
- Proyecto Asociado

id = SHA3("15-10-23::14:02:23" + "DC56RT89" + "152.50" + "PY-100")

Cuando se tenga el hash de cada bloque de información, se genera un nuevo hash con el mismo método pero utilizando los hash de los anteriores con los nodos internos, ejemplo

id = SHA3("98366e7eca8...." + "a41602563256284....")



Luego de generar este árbol, se tomará el valor del hash Final que sería el root_merkle y este será guardado en un listado de bloques, simulando un sistema de blockchain el cual tiene una similitud a una lista doblemente enlazada, cada bloque guardará la siguiente información:

- **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).
- **Data:** Esta data será el merkle root
- **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, data.

Dependencias de tareas:

Este apartado tiene como finalidad utilizar la lista de tareas de antecesores y sucesores de la primera fase del proyecto, por lo cual se creará un archivo con extensión .json, el cual tendrá un el código de tarea, y las tareas que depende de ella para poder realizarla, se tendrá 4 atributos en cada una de las tareas, la cuales son el proyecto, código de la tarea, sus antecesores y el pago asignado a la tarea. Si el arreglo no posee ningún elemento, significa que la tarea no necesita de ninguna otra previa para poder comenzar, para esto se muestra el siguiente ejemplo. **Para esto se usará el grafo dirigido mediante una lista de adyacencia para su mayor manipulación.**

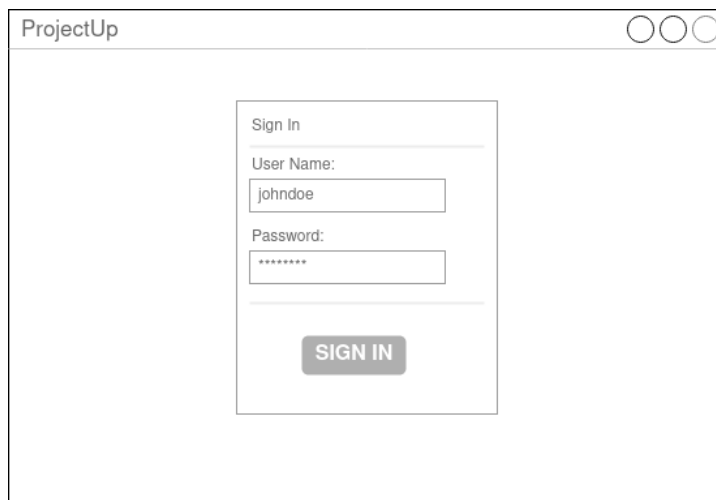
```
{
  "Tareas": [
    {
      "proyecto": "PY-100",
      "codigo": "T1-PY-100",
      "antecesor": [],
      "pago": "152"
    },
    {
      "proyecto": "PY-100",
      "codigo": "T2-PY-100",
      "antecesor": [],
      "pago": "162"
    },
    {
      "proyecto": "PY-100",
      "codigo": "T3-PY-100",
      "antecesor": [
        {
          "codigo": "T2-PY-100"
        },
        {
          "codigo": "T1-PY-100"
        }
      ],
      "pago": "222"
    },
    {
      "proyecto": "PY-100",
      "codigo": "T4-PY-100",
      "antecesor": [
        {
          "codigo": "T3-PY-100"
        }
      ],
      "pago": "125"
    }
  ]
}
```

Usuario de Empleado (FDEV, BDEV y QA)

Para todos los empleados que no sean Project Manager y que estén registrados en el sistema se les dará acceso a las siguientes funcionalidades:

Iniciar Sesión:

Los usuarios registrados tendrán que ingresar el ID de empleado como usuario y la contraseña previamente cargada del archivo CSV de empleados. De ingresar los datos incorrectos o que no exista el usuario se deberá de mostrar un mensaje de error en la interfaz de usuario.



The screenshot shows a window titled "ProjectUp" with a standard macOS-style title bar (three circles). In the center is a "Sign In" form. The form has a title "Sign In", a "User Name:" label, a text input field containing "johndoe", a "Password:" label, a password input field with masked characters "*****", and a "SIGN IN" button at the bottom.

Ver sus tareas:

En la ventana principal o de inicio se le mostrarán todas las actividades que estén asignadas al usuario logueado. Deberán de realizar un recorrido de su árbol B y almacenar dentro de una **Lista Simple** todas la tareas que concuerden con el id del usuario en búsqueda. Esta lista simple será para mostrar las actividades dentro de su interfaz.



The screenshot shows the main interface of the "ProjectUp" application. At the top right, it displays the user ID "BDEV-452" next to a user profile icon. Below this is a "Tareas" tab. The main content area features a "Filtrar" dropdown menu with a downward arrow, and the text "Filtro: Sin ningún filtro". Below the filter is a table titled "Tareas: 1".

Código de tarea	Nombre de proyecto	Nombre de tarea
T1-PY-100	Sistema de inventario Vestidos Julieta	Generar consultas del area de destornilladores

Estado de las Tareas:

A partir de la lista de tareas, el empleado podrá cambiar los estados de las tareas, para lo cual existen 3 estados posibles:

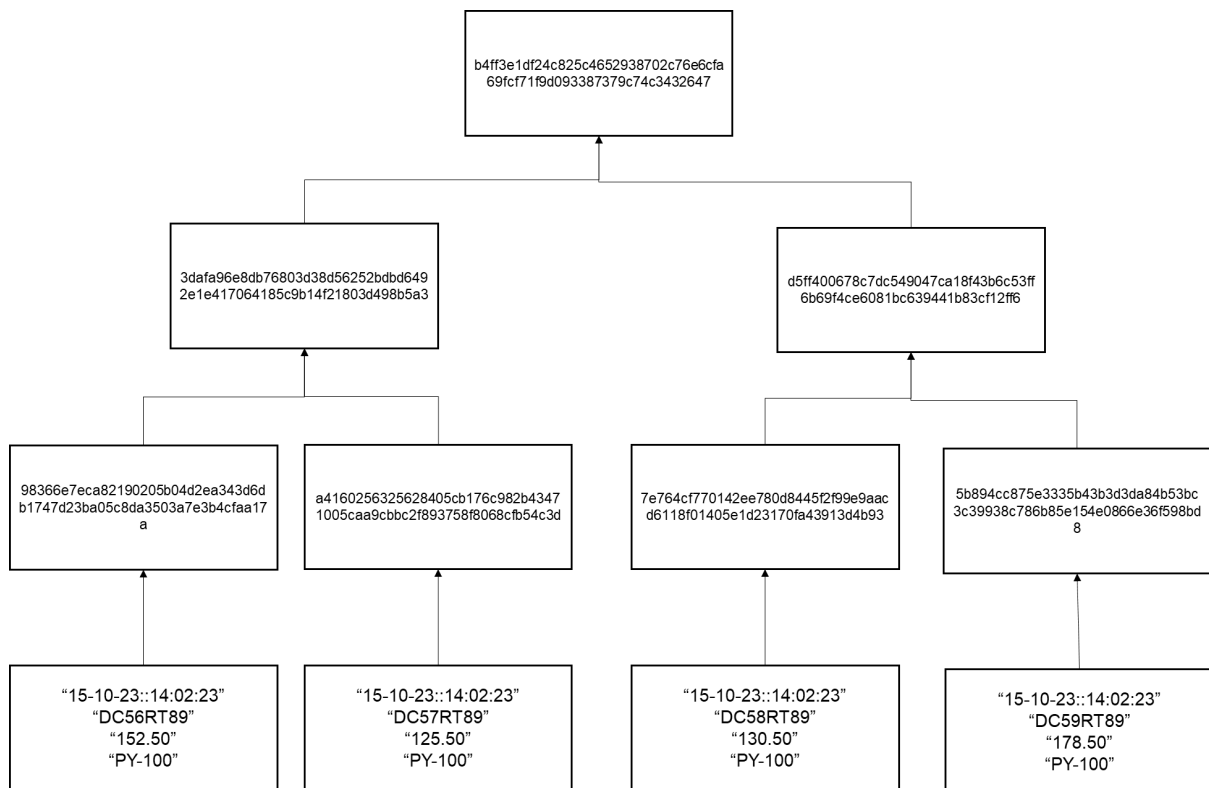
- Estado 1: "Sin comenzar", Representa que la tarea no se ha comenzado
- Estado 2: "En proceso", Representa que la tarea ya se está realizando, para esto debe verificarse que su tarea antecesora o dependiente ya está completada (Estado 3) de lo contrario debe alertar al empleado que no puede cambiar a este estado hasta que la o las tareas dependientes ya estén completas.
- Estado 3: "Terminado", Representa que la tarea ya se finalizó con éxito.

Área de Reportes

Se contará con un apartado especial en el que se puedan mostrar en forma de grafos las estructuras en tiempo real, generando la imagen utilizando el visualizador de imágenes del sistema operativo. Los reportes sólo serán generados con la herramienta de Graphviz.

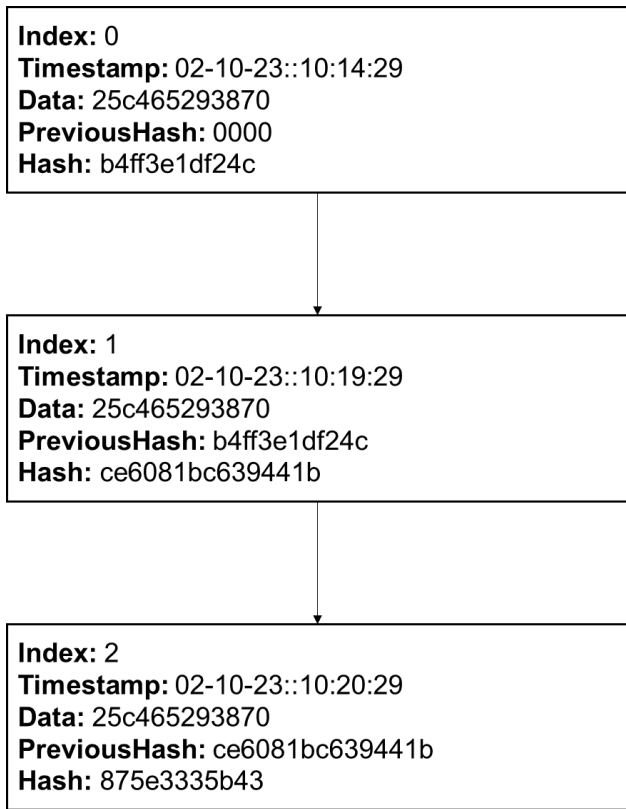
Reporte de Árbol de Merkle:

Para la realización del reporte se hará de la siguiente manera, mostrando únicamente las cadenas resultantes de las funciones hash a excepción de la última línea que muestra los datos que se utilizó para esas funciones. Este reporte se genera solo una vez con el nombre, arbolMerkle-#proyecto, ejemplo: arbolMerkle-100.



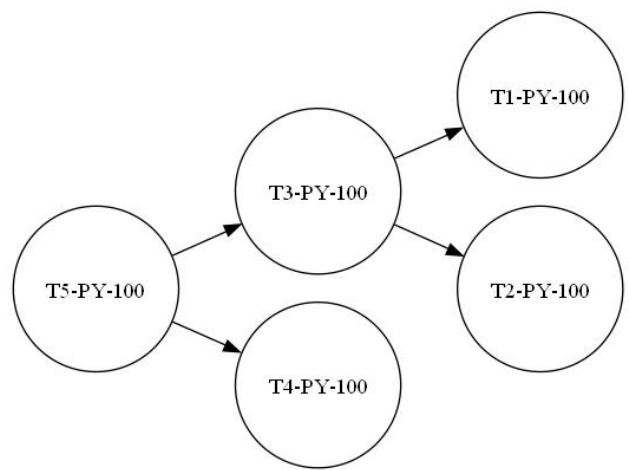
Sistema de Blockchain:

Para la realización del reporte se hará de la siguiente manera, todos los atributos de la siguiente manera.



Reporte de Grafo:

Para la realización del reporte se usará la lista de adyacencia para partir del grafo, para ello se mostrará el código de tarea y su dependencia de la siguiente manera:



Manuales

(Técnico y Usuario)

Manual de Usuario:

Para el manual de usuario se deberán tomar capturas de cada pantalla que se utilice dentro de la app e indicar cual es la funcionalidad de cada botón o tabla que tenga el acceso al usuario. En otras palabras, una guía para el usuario que utilizará su aplicación, por lo tanto debe de tener los siguientes aspectos:

- Capturas de pantalla de vistas de usuario
- Indicar acción de cada botón de la interfaz
- Enumerar las funcionalidades que posee cada pantalla.

Manual Técnico:

El manual técnico, se deberán de describir aspectos lógicos, librerías y todos los componentes que hacen funcionar su aplicación, por lo que su manual técnico deberá de poseer los siguiente:

- Librerías utilizadas en el desarrollo
- Explicación de las estructuras de dato utilizadas en el proyecto
- Enumerar funciones, métodos y constructores de cada clase con una breve explicación de su funcionamiento.

Tecnologías a utilizar

Realizar la aplicación en Python:

Este contará con una aplicación con interfaz gráfica realizada también en el mismo lenguaje que las estructuras anteriormente mencionadas,

Para interfaz gráfica Figma o Tkinter:

Para la creación de interfaces gráficas dentro del proyecto queda a discreción del estudiante cuál implementar, las herramientas de más fácil uso son Figma y Tkinter.

Realizar los reportes en Graphviz:

Todos los reportes deberán estar realizados en graphviz y deben estar constantemente generando cuando se realice un cambio, para poder observar de forma visual el estado actual de las estructuras.

Restricciones

- Las estructuras deben de ser desarrolladas por los estudiantes sin el uso de ninguna librería o estructura predefinida en el lenguaje a utilizar.
- 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 y aquellas secciones del proyecto que estén relacionadas.
- Los manuales sólo se deberán realizar en formato Markdown o .md, por lo que cualquier otro formato se anularán los puntos de esa área.

Observaciones

- El lenguaje para esta fase será Python.
- Herramienta de desarrollo de reportes solamente **Graphviz**.
- La entrega se realizará por medio de GitLab, el nombre del repositorio debe ser **EDD_2S2023_PY_#carnet**, donde se creará una carpeta con el nombre **EDD_PY1_Fase3**. Y por medio de **UEDI** se hará entrega del link de su repositorio.
- Recordar tener sus repositorios en privado, para evitar copias de código.
- Deben de agregar a los auxiliares como colaboradores en los repositorios según su sección:
 - **Sección A: 2636889760101**
 - **Sección B: leonardo0martinez**
 - **Sección C: CristianMejia2198**
- Realizar el **manual de técnico y manual de usuario en archivos markdown diferentes**, en el README colocar solo link a cada manual 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: **2 de Noviembre, a las 18:00 horas**.
- Las copias encontradas serán penalizadas con 0 y reportadas a la Escuela de Ciencias y Sistemas.