

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



Ingenieros:

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

Auxiliares:

- Aldo Perez
- Leonardo Martínez
- Cristian Suy

EDD ProjectUp

FASE II

Ciudad de Guatemala, 01 de septiembre 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, eliminación e inserción para árboles y algoritmos de hashing para tablas hash.
- Desarrollar interfaces gráficas en Python.

Resumen de estructuras a utilizar:

- **Árbol AVL:** Almacenamiento de proyectos.
- **Árbol B:** Tareas de todos los proyectos.
- **Tabla Hash:** Lista de Empleados General.

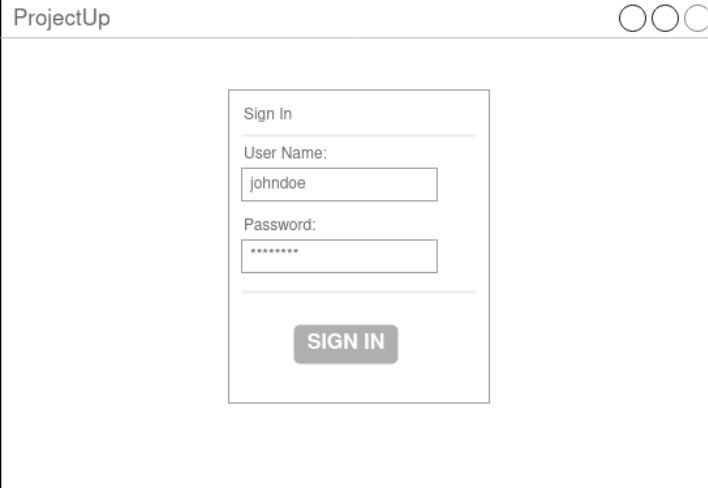
Definición del problema

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 screenshot shows a web application window titled "ProjectUp" with standard window controls (minimize, maximize, close) in the top right corner. Centered within the window is a "Sign In" form. The form contains a "User Name:" label followed by a text input field containing the text "johndoe". Below this is a "Password:" label followed by a password input field filled with asterisks. At the bottom of the form is a grey button labeled "SIGN IN".

Carga Masiva de Empleados:

Los tipos de empleados que se utilizarán dentro de la app son los siguientes (Todos los tipos de usuario a excepción del Project Manager se le podrán asignar tareas):

- Frontend Developer
- Backend Developer
- Quality Assurance (QA)
- Project Manager

El Project Manager también tendrá un apartado en el que podrá subir un archivo CSV en el que contenga una lista de empleados para agregar al registro de la **Tabla Hash**, quedando a discreción del estudiante que librerías utilizar para la lectura del archivo. La estructura del archivo sería la siguiente:

	A	B	C	D
1	id	nombre	contraseña	puesto
2	FDEV-001	Leonardo Martinez	leo1234	Frontend Developer
3	BDEV-002	Cristian Suy	cris1234	Backend Developer
4	QA-003	Ebany Larios	qwerty123	Quality Assurance
5	FDEV-004	Carlos Maldonado	car123	Frontend Developer
6	BDEV-005	Juan Perez	juan123	Backend Developer
7	QA-006	Diego Hernandez	eldiego	Quality Assurance
8	FDEV-007	Emy Lopez	emy123	Frontend Developer
9	BDEV-008	Esteban Sosa	este1234	Backend Developer
10	QA-009	Kevin Mendoza	kev120	Quality Assurance

Para la tabla hash de empleados comenzará con 5 espacios disponibles, luego de llegar al 70% de utilización, aumentará la capacidad según la serie de Fibonacci iniciando a partir del valor de 5. Cuando llegue al 70% de utilización, el nuevo tamaño de la tabla hash será 8, 13, 21 y así sucesivamente.

Para la función de inserción a la tabla hash será el código del empleado, lo cual necesitará tomar cada carácter del código para convertirlo a su código ASCII y sumarlo de la siguiente manera: si el código es BDEV-001 daría el siguiente arreglo [66, 68, 69, 86, 45, 48, 48, 49], sumando daría un total de 479, ese número se pasará como parámetro a 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.

ProjectUp

PM-202300795



Empleados

...

Carga masiva

Empleados: 1

Código	Nombre	Tipo de empleado
BDEV-001	Cristian Suy	Backend Developer

Carga de Archivo JSON:

El project manager usará un archivo con extensión .json para realizar la carga de tareas en la aplicación, este archivo contará con información clave el cual será utilizado para los empleados. Tomar en cuenta que el archivo JSON tendrá información clave tal como, código o ID del proyecto, nombre del proyecto, las tareas que deben realizar en cada uno de los proyectos así como sus encargados, y la prioridad del mismo.

El archivo JSON tendrá la siguiente estructura:

```
{
  "Proyectos":[
    {
      "id":"PY-100",
      "nombre":"Sistema de inventario Vestidos Julieta",
      "tareas": [
        {
          "nombre":"Generar Consultas del area de destornilladores",
          "empleado":"BDEV-452"
        },
        {
          "nombre":"Mostrar Herramientas de trabajo",
          "empleado":"FDEV-414"
        },
        {
          "nombre":"Probar barra de busqueda",
          "empleado":"QA-123"
        }
      ],
      "prioridad":"B"
    },
    {
      "id":"PY-101",
      "nombre":"Tienda Online Restaurante Giovanni",
      "tareas":[],
      "prioridad":"B"
    },
    {
      "id":"PY-102",
      "nombre":"Sistema de inventario EPA",
      "tareas":[],
      "prioridad":"A"
    }
  ]
}
```

Toda esta información se estará almacenando en diferentes estructuras las cuales explicaremos a continuación:

Almacenamiento de Proyectos:

Los almacenamientos de proyectos debe dar lectura al archivo JSON, los atributos necesarios para este apartado son “id” (Id de Proyecto), “nombre” (Nombre del proyecto) y “prioridad”, los cuales serán almacenados en un **Árbol AVL** tomando como referencia el id del proyecto para la inserción a dicha estructura.

Adicionalmente a eso, cada nodo de la estructura debe guardar mínimo la siguiente información:

- ID del proyecto
- Nombre del Proyecto
- Prioridad

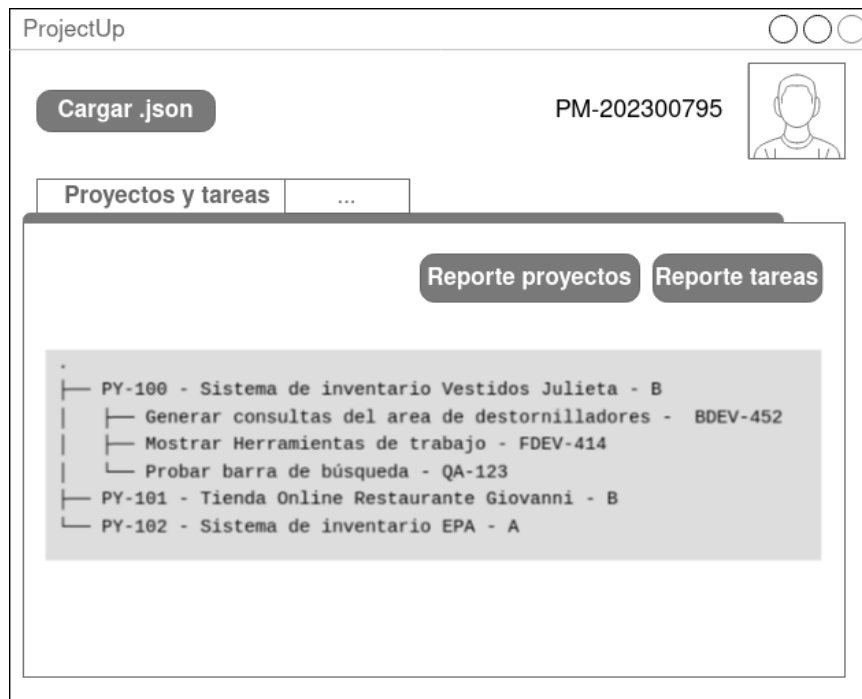
Almacenamiento de tareas:

Para el almacenamiento de tareas se hará de manera global, para esto es importante tomar en cuenta los siguientes atributos del archivo json, “**id**” (Id de Proyecto) y “**tareas**” (Array de Tareas) el cual posee los atributos “**nombre**” (nombre de la tarea) y “**empleado**” (Id del empleado a cargo), para el almacenamiento, por motivos de eficiencia, se utilizará el **Árbol B de grado 3**, antes de realizar la inserción debe generarse un ID para cada tarea ya que ese será la referencia para insertar a dicho árbol, para generar un ID a cada tarea se usará la siguiente estructura, se creará colocando “**T**”+ **Número de Tarea** +“-”+ **Id de Proyecto**, por ejemplo, un proyecto con el id PY-100 posee 3 tareas, entonces se genera los siguientes códigos:

- T1-PY-100
- T2-PY-100
- T3-PY-100

Adicionalmente a eso, cada nodo de la estructura debe guardar mínimo la siguiente información:

- Código de Tarea (explicado anteriormente)
- Nombre de Tarea
- Código de Encargado
- Código de Proyecto (de ser necesario)

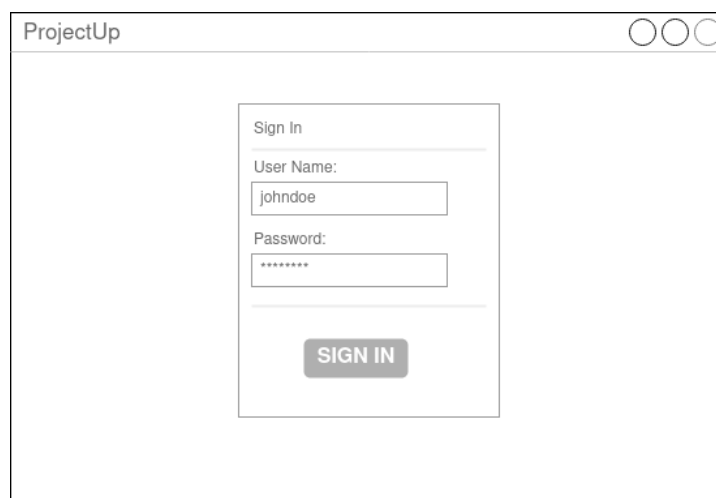


Usuario de Empleado (FDEV, BDEV y QA)

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

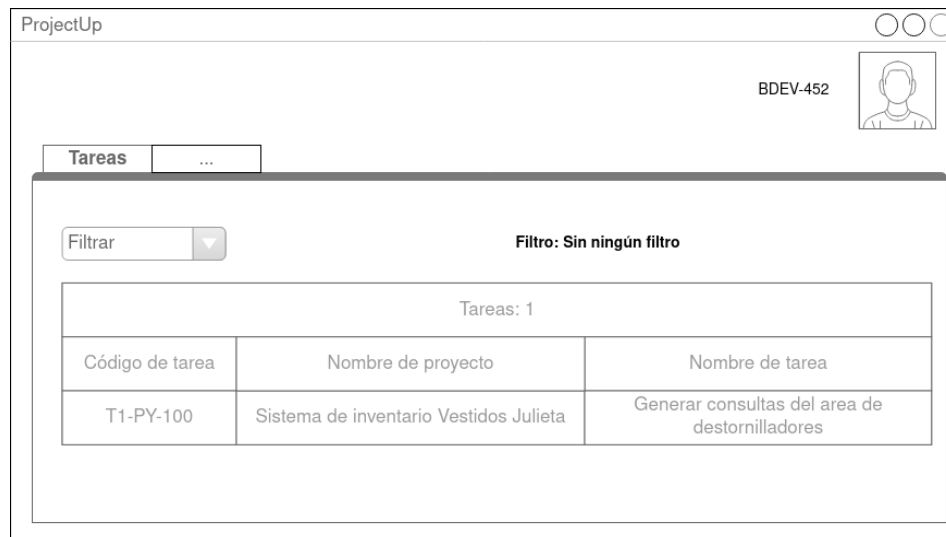
Iniciar Sesión:

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.



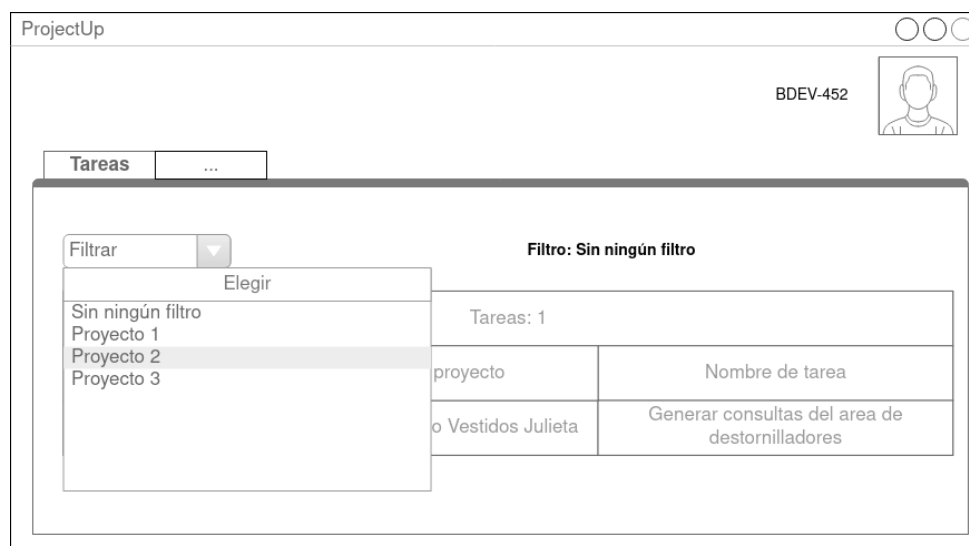
Ver sus tareas:

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.



Filtros de Tareas:

A partir de la lista de tareas, el empleado podrá filtrar las tareas de acuerdo a los proyectos en dónde el empleado posea asignadas las tareas. Deberá de implementar un apartado gráfico para la selección de los proyectos que esté activo o mostrar todas sus tareas y a qué proyecto pertenece cada una.



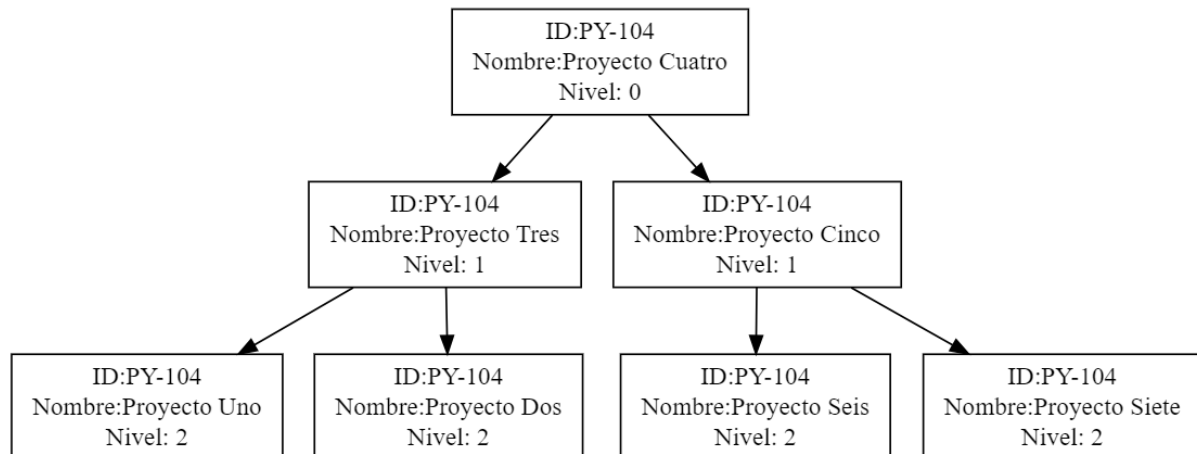
Área de Reportes

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.

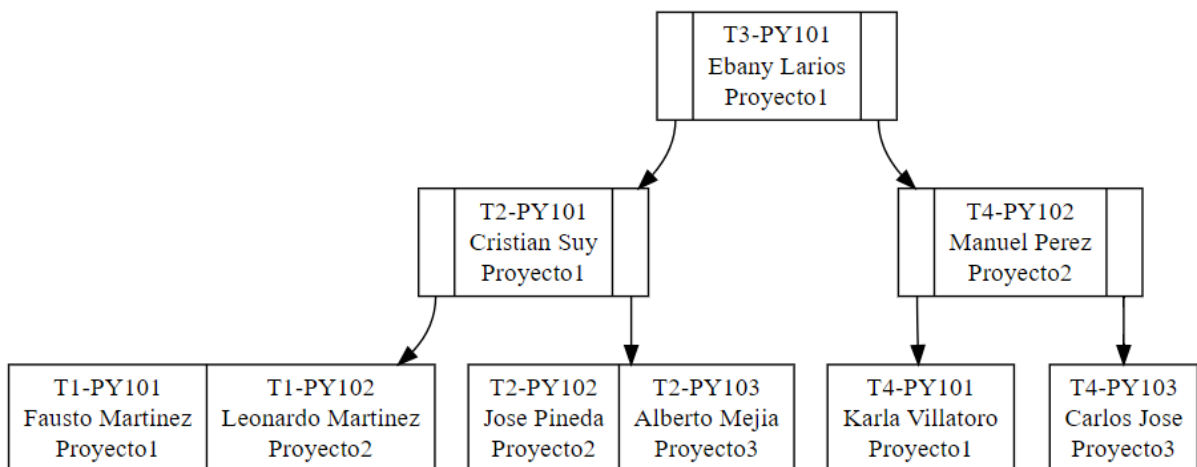
Reporte del Árbol AVL:

Para la realización del reporte se hará de la siguiente manera, mostrando nada más el código del proyecto.



Reporte del Árbol B:

Para la realización del reporte se hará de la siguiente manera, mostrando el código de la tarea, encargado y nombre de proyecto.



Reporte de Tabla Hash:

Será la tabla de empleados de la interfaz, para el usuario administrador. Se deberá de colocar el index que fue generado al momento de la inserción del empleado en una **columna extra de la tabla**.

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 en consola en el que se deberán de retornar los datos en un formato JSON, se puede utilizar cualquier librería para poder crear el archivo, esto para utilizarlo como base para las fases posteriores del proyecto.

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_Fase2**. 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 Octubre, a las 23:59 horas**.
- Las copias encontradas serán penalizadas con 0 y reportadas a la Escuela de Ciencias y Sistemas.