Proyecto → Fase 1

SMART CLASS

Ing. Jesus Guzman, Ing. Alvaro Hernandez, Ing. Luis Espino

Randolph Muy, Jorge Salazar, Josué Pérez, José Véliz

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Estructuras de Datos

Visión General

A raíz de la situación por la cual atraviesa la sociedad, la digitalización ha sido un total apoyo a casi todas las actividades que se realizan, por lo cual se le solicita a usted como desarrollador, la creación de una aplicación de apoyo a la universidad haciendo uso de los conocimientos y capacidades que ha adquirido hasta el punto en el cual se encuentra en la carrera.

Smart class será una plataforma diseñada para el apoyo a los estudiantes universitarios a organizar de mejor manera las actividades que realizan en su dia a dia, esto para poder manejar de manera ordenada las mismas, lo cual facilitará la visualización y administración de estas actividades, esta aplicación será capaz de almacenar no solamente las tareas sino también los estudiantes que forman parte de la misma, todos estos datos deben ser analizados y almacenados de manera segura y confiable.

Objetivos

- Que el estudiante se familiarice con el lenguaje de programación C++.
- Que el estudiante sepa envolverse en el ámbito del manejo de la memoria.
- Comprender la diferencia entre memoria estática y dinámica.
- Familiarizarse con el uso de Git.
- Que el estudiante se familiarice con el manejo de lectura de archivos.
- Comprender el uso de arreglos multidimensionales.
- Calcular posiciones de memoria haciendo uso de Col Major y Row Major.

Especificaciones

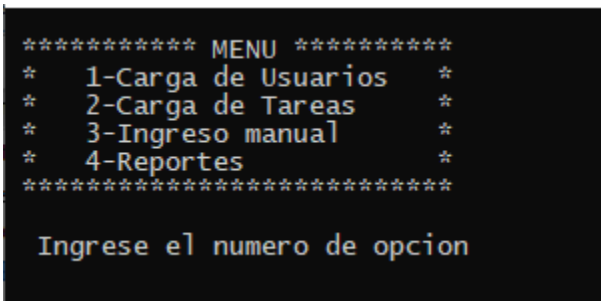
Para esta fase de proyecto se solicita crear una aplicación de consola desarrollada en el lenguaje C++ . En esta parte se cargarán distintos archivos de entrada con los datos para realizar una limpieza y análisis de estos, con los datos analizados se crearán distintos archivos de salida que serán utilizados en fases posteriores de creación del proyecto.

Menú

Debido a que el desarrollo de la aplicación es en consola utilizando el lenguaje de programación C++, es necesario contar con una menú para realizar todas las funcionalidades que la aplicación requiera para su correcto funcionamiento.

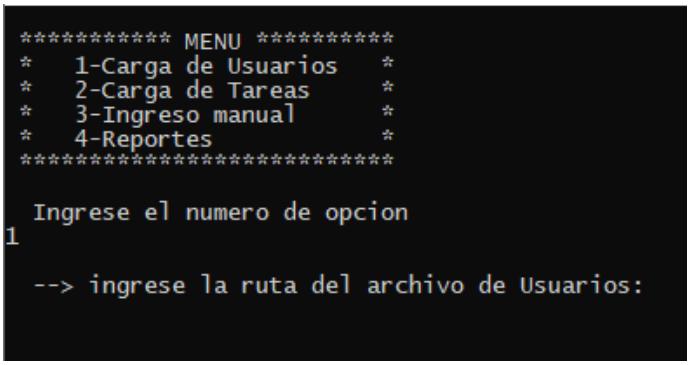
A continuación se listan las funcionalidades principales, y queda a discreción del estudiante agregar más funcionalidades que considere necesarias.

- Carga de estudiantes
- Carga de tareas
- Ingreso manual
- Reportes



Carga de estudiantes

Al momento de seleccionar la opción “Carga de estudiantes” se desplegará un opción en donde se solicitará la ruta ó el Path del archivo de entrada, lo cual al momento de presionar enter, la funcionalidad leerá la información del archivo y se procederá a realizar la acción de carga.



Carga de tareas

Al momento de seleccionar la opción “Carga de Tareas” se desplegará un opción en donde se solicitará la ruta ó el Path del archivo de entrada, lo cual al momento de presionar enter, la funcionalidad leerá la información del archivo y se procederá a realizar la acción de carga.

Ingreso Manual

Al momento de seleccionar la opción “Ingreso Manual” se desplegará un submenú, el cual tiene como finalidad contar con funcionalidades como el ingreso de estudiantes y tareas de manera **manual**, de modo que al seleccionar “**estudiantes**” o “**Tareas**” se desplegará **otro** submenú el cual contará con la opción de **Ingresar**, **Modificar** o **eliminar**, con respecto a estudiantes o tareas (Más adelante se detalla con mejor claridad cada opción) .

Al momento de escoger la opción de salir, se retorna al menú anterior, hasta llegar al principal.

```
********** MENU **********
*   1-Carga de Usuarios   *
*   2-Carga de Tareas     *
*   3-Ingreso manual      *
*   4-Reportes            *
*************************

 Ingrese el numero de opcion  3

********** MENU **********
*   1-Usuarios            *
*   2-Tareas              *
*   3-salir               *
*************************

 Ingrese el numero de opcion  1

****** MENU USUARIOS ******
*   1-Ingresar            *
*   2-Modificar           *
*   3-Eliminar            *
*   4-Salir               *
*************************

 Ingrese el numero de opcion
```

Reportes

Al momento de seleccionar la opción “Reportes” se desplegará un submenú, el cual tiene como finalidad contar con funcionalidades como el reporte de la lista de estudiantes y la linealización de tareas. Por lo tanto al escoger cualquiera de las opciones se debe proceder a graficar la respectiva estructura, haciendo uso de la herramienta **Graphviz**, y mostrar el respectivo mensaje de error o notificación. El menú de reportes también debe contar una opción de **salir**, para regresar al menú principal.

```
********** MENU **********
*   1-Carga de Usuarios   *
*   2-Carga de Tareas     *
*   3-Ingreso manual      *
*   4-Reportes            *
*************************

 Ingrese el numero de opcion  4

******* MENU REPORTES*******
*   1-Lista Usuarios      *
*   2-Linealizacion Tareas *
***************************

 Ingrese el numero de opcion
```

Estudiantes

Smart class está enfocado a estudiantes universitarios, los cuales serán los usuarios de esta aplicación y estos se almacenarán en una **lista doblemente enlazada circular.** Los datos a almacenar de cada estudiante serán los siguientes:

- Número de carnet
- DPI
- Nombre
- Carrera
- Correo
- Password
- Créditos
- Edad

*Se deberá validar que los datos en cuanto a DPI y carnet cumplan con la cantidad de dígitos correspondiente a cada uno (DPI **13 dígitos**, carnet **9 dígitos**), de igual manera el correo debe cumplir con la estructura de un correo válido (**user@dominio.[com|es|org]**).

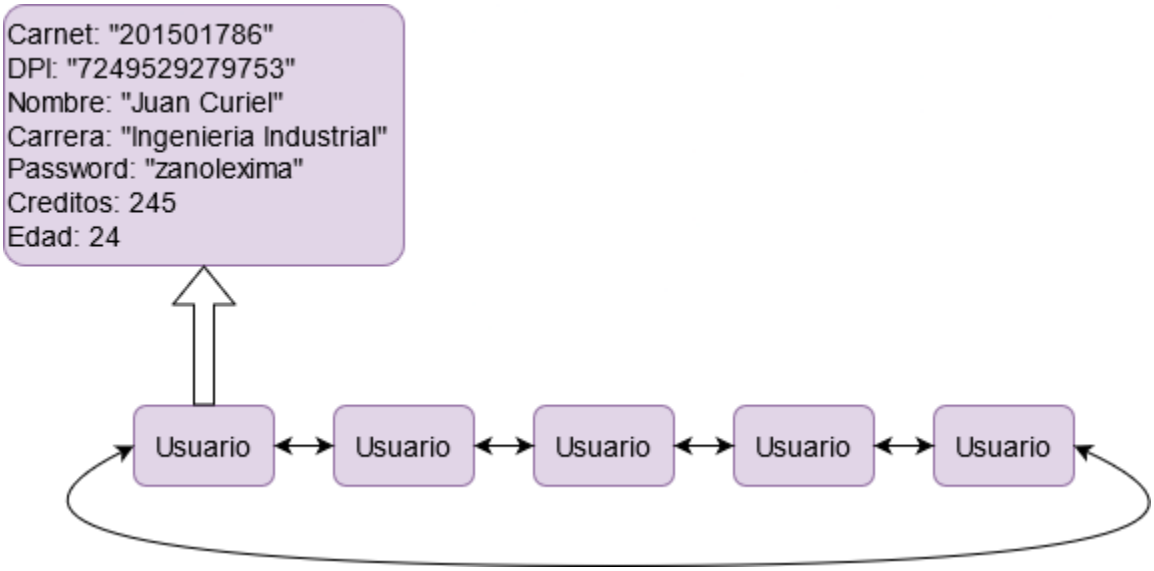
De encontrarse un error se deberá reportar en la cola de errores con la identificación adecuada del mismo, esto será mencionado más adelante.

Carga Masiva de estudiantes

Los datos de los estudiantes serán cargados mediante un archivo el cual tendrá el siguiente formato **CSV .**

Ejemplo:

Estudiantes.csv



Operaciones sobre estudiantes

El sistema permitirá **agregar, modificar y eliminar** estudiantes, ingresando los datos mediante consola, estas operaciones se describen a continuación:

- **Agregar un estudiante:** Luego de realizar la carga masiva correspondiente es posible querer agregar un nuevo estudiantes, esto será posible por medio del menú correspondiente, en donde al ingresar a la opción de agregar le solicitará todos los datos de un estudiante especificados anteriormente.
- **Modificar un estudiante:** Es posible modificar los valores de un estudiante ya insertado, para ello al entrar en el menú correspondiente se solicitará el DPI del estudiante, al ingresar el DPI se visualizará una serie de opciones correspondientes a todos los valores de un estudiante, y en donde la persona puede elegir qué valor desea alterar o cambiar.
- **Eliminar un estudiante:** Otra operación que se puede realizar es la eliminación de estudiantes en este caso si un estudiante se desea eliminar se debe localizar en el menú correspondiente e ingresar el DPI del estudiante a eliminar, se debe mostrar un mensaje de advertencia para validar que si se desea eliminar ese estudiante, si se acepta el estudiante será eliminada de la lista.

Tareas

Otro aspecto importante que maneja Smart Class son las tareas, este apartado será el encargado de registrar todas las tareas que desea almacenar un estudiante, para ello se deben manejar los siguientes datos por cada tarea creada:

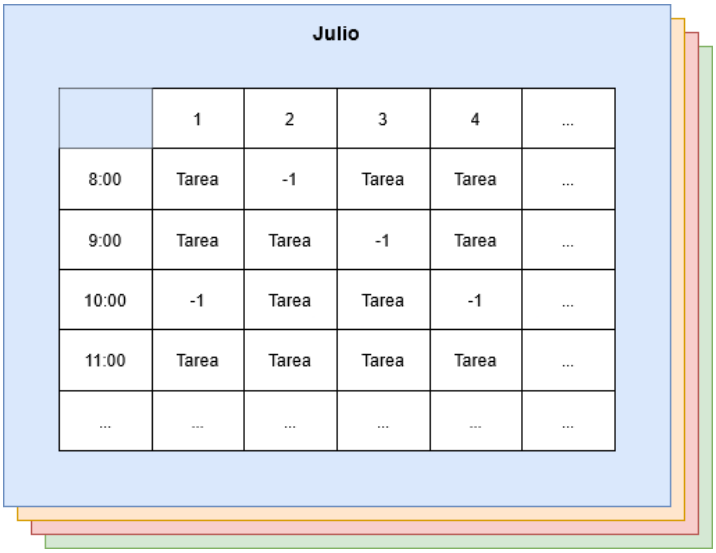
- Id de tarea(Secuencial)
- Carnet
- Nombre de la tarea
- Descripción
- Materia
- Fecha
- Hora
- Estado(Pendiente, Realizado, Incumplido)

*Se deberá **validar que el carnet colocado en la tarea sea correspondiente a alguno de los carnet ingresados en la lista de estudiantes** mencionada en el apartado anterior, de igual manera se deberá validar que las fechas sean coherentes en cuanto a **días y meses** y que el formato de ingreso de las mismas sea **YYYY/MM/DD** (los meses que se utilizarán contarán todos con 30 días), para convertirlo a un formato diferente que será mencionado más adelante para el archivo de salida, así mismo también **validar que la hora ingresada sea coherente** (no estar fuera del rango definido en el siguiente apartado), de la misma manera que en el apartado anterior, los errores encontrados se deberán agregar a la cola de errores con la identificación adecuada del mismo, esto será mencionado más adelante.

Carga Masiva de Tareas

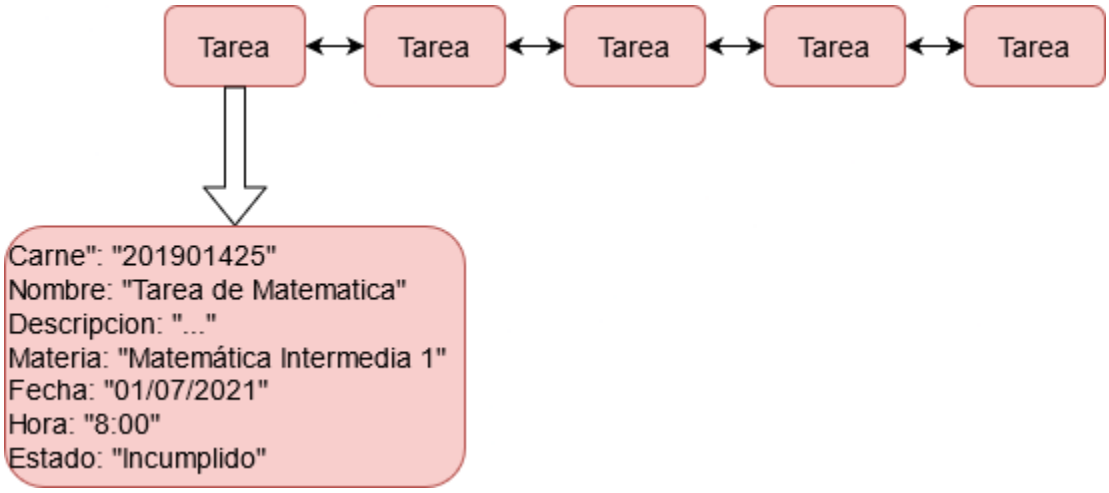
Las tareas que maneja el sistema serán cargadas por medio de un archivo **CSV**, el cual tendrá una estructura de una matriz de tres dimensiones, las dimensiones serán las siguientes:

1. Meses
2. Días
3. Horas



Ejemplo:

Tareas.csv



El proceso que debe seguir la carga de las tareas será mostrado a continuación, detallando el paso a paso que se debe seguir para poder ingresar las tareas desde el archivo de carga hasta la estructura deseada, siendo en este caso un **Lista Doblemente Enlazada**.

1. En primer lugar se debe leer el archivo CSV(queda a discreción del estudiante el método para leer el archivo), esto se hará por medio de la ruta del archivo.
2. Luego cuando el archivo ya se encuentra leído, se debe proceder a almacenarse en un arreglo de tres dimensiones estático, para poder cumplir con esto solo será posible tener

una tarea por una hora. En caso de encontrar valores faltantes se debe llenar la información con valores nulos, en este caso con el valor **-1**.

3. Cuando el arreglo estático esté cargado se debe proceder a linealizar la información por medio de Row Major o Column Major(Consulte las consideraciones en el área de abajo).
4. En el momento que se realiza la linealización se debe proceder a almacenarse en una estructura dinámica siendo en este caso una Lista doblemente enlazada.

Algo importante a tener en cuenta es que este proyecto se encuentra en una fase inicial, de tal manera que se trabajarán con datos limitados para poder realizar una prueba del funcionamiento del mismo, es por ello que las tareas se encontrarán establecidas únicamente dentro del semestre actual, respetando la siguiente tabla:

Fecha	Rango válido
Meses	De Julio hasta Noviembre
Días	Todos los días del mes correspondiente
Horas	8:00 - 16:00

Operaciones sobre Tareas

Luego de la realización de la carga masiva existirán una serie de operaciones que se pueden realizar sobre la Cola resultante del proceso de carga de las tareas, estas operaciones se describen a continuación:

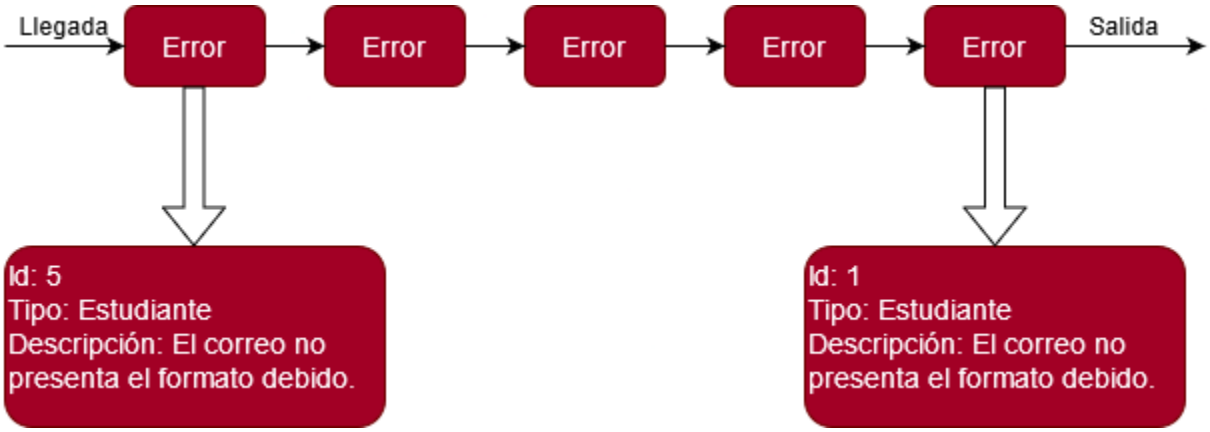
- **Ingresar una tarea:** Luego de la carga masiva es posible querer ingresar una nueva tarea, y esto se logrará por medio del menú correspondiente, en donde al seleccionar la opción de ingresar solicitará todos los datos de una tarea, exceptuando el Id de la tarea que será ingresado de forma automática. Se deberá calcular la posición linealizada donde deberá agregarse la tarea y de no estar ocupada se almacenará en esa posición.
- **Modificar una tarea:** Es posible modificar una tarea ya insertada, para ello al entrar en el menú correspondiente se solicitará el indice de la tarea, al ingresarlo mostrará una serie de opciones correspondientes a todos los valores de la tarea, exceptuando el Id de la tarea que no se debe poder cambiar.
- **Eliminar una tarea:** Otra operación que se puede realizar es la eliminación de tareas en este caso si un estudiante desea eliminar una tarea debe localizarse en el menú correspondiente e ingresar el índice de la tarea a eliminar, se debe mostrar un mensaje de advertencia para validar que si se desea eliminar esa tarea, si se acepta la tarea será eliminada de la lista.

Errores

Para el manejo de los errores de los datos ingresados a la aplicación se tomarán en cuenta las consideraciones mencionadas en los apartados anteriores, siendo los elementos de cada error los siguientes, todos estos deben ser almacenados en una estructura tipo **COLA**:

- Id de error (valor autoincremental).
- Tipo (Estudiante o Tarea).
- Campo que no cumple con especificaciones (descripción de error).

Para cada error **se debe usar la opción de modificar el elemento afectado** para poder proceder a generar el archivo de salida, **de lo contrario el archivo de salida no podrá ser generado** y se mostrará un error que aún faltan elementos por arreglar.



Reportes

Luego de realizar la carga masiva se habilitará la opción de obtener reportes sobre los datos ya ingresados, como fue posible visualizar con anterioridad en el ejemplo del menú. existen en su totalidad cuatro reportes, los cuales son:

1. Reporte sobre la lista de estudiantes
2. Reporte sobre la lista de tareas linealizadas
3. Búsqueda en estructura linealizada.
4. Búsqueda de posición en lista linealizada
5. Cola de Errores
6. Código generado de salida

En el momento de elegir uno de los dos primeros reportes debe ser capaz de generar la gráfica correspondiente y almacenarla(se deja a discreción del estudiante el lugar de almacenamiento, solamente procurando que sea fácil de acceder), esta imagen debe de estar en formato png, además si se solicita que se genere una nueva imagen esta debe de ser nueva y no sustituir la anterior, esto permitirá la verificación de que los cambios se hayan realizado.

El tercer reporte consiste en un método dentro de la lista doble linealizada, este método permitirá realizar una búsqueda de una tarea por medio del ingreso los tres parámetros necesarios, para ello le será solicitado al estudiante que ingrese el mes, día y hora de la tarea, y con esos parámetros se realizará la búsqueda de la tarea, devolviendo como tal la información de la tarea si es que existe, o en caso contrario indicando por medio de un mensaje que la tarea no existe dentro de la lista doble.

El cuarto reporte consiste en ingresar por medio de consola tres parámetros relacionados con la matriz de tres dimensiones, estos tres parámetros serán el mes, el día y la hora, con estos datos el estudiante debe calcular la posición en la cual se encontraría este registro dentro de la lista enlazada e imprimirlo en consola como respuesta de esa búsqueda solicitada.

El quinto reporte consiste en generar un archivo en formato png el cual debe mostrar una imagen de una cola con los errores encontrados al momento de leer los archivos de entrada.

El sexto reporte consiste en generar un archivo con un código especial, este archivo permitirá almacenar todos los datos recolectados y servirá de entrada para las próximas versiones de nuestro proyecto, para poder comprender cómo se encuentra estructurado se incluye al final un ejemplo del mismo. Algo importante a considerar es lo siguiente, primero: la extensión del archivo debe ser *.txt, segundo el formato de la fecha de salida debe ser el siguiente

DD/MM/YYYY, y por último, se deja a discreción del estudiante el lugar de almacenamiento, solamente procurando que sea fácil de acceder.

Ejemplo:

Estudiantes.txt

Consideraciones

1. El lenguaje a utilizar será C++.
2. Las estructuras serán realizadas por el estudiante.
3. IDE a utilizar es libre (Recomendaciones: Codeblocks, Dev C++, Visual Studio Code).
4. El tipo de linealización es dependiente de su carné, los carné impares deben realizarlo por medio de Column Major y los carné par la deben realizar por Row Major.
5. La entrega será por medio de la plataforma UEDI.
6. El estudiante debe tener un repositorio privado en github con el nombre **EDD_SmartClass_#Carné** y agregar a su tutor como colaborador al repositorio del proyecto (Cada tutor les hará llegar su estudiante).
7. Será calificado del último commit realizado antes de la fecha y hora de entrega.
8. Las copias totales o parciales **serán penalizadas con nota de 0 puntos y reportadas** ante la escuela de ciencias y sistemas.
9. **Se deberán graficar todas las estructuras de la práctica con graphviz**.
10. **Fecha de entrega: 21 de agosto de 2021 antes de las 23:59 horas.**