

## Diseño

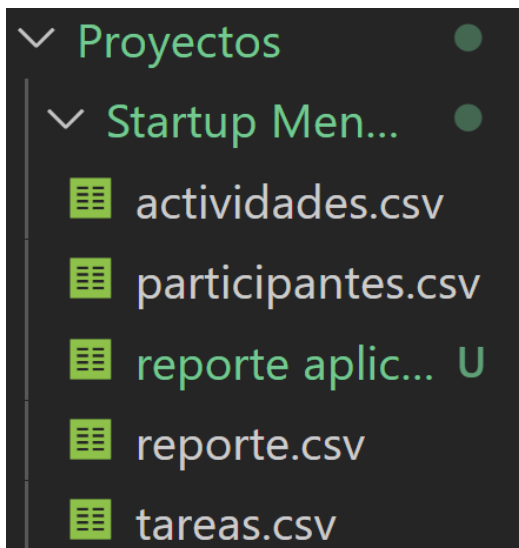
**Sergio Franco Pineda, Juan Pablo Rodríguez Briceño, Juan Pablo Martínez Pineda**

### Enunciado

Se desea construir una aplicación que permita manejar y registrar el avance de un proyecto en equipo.

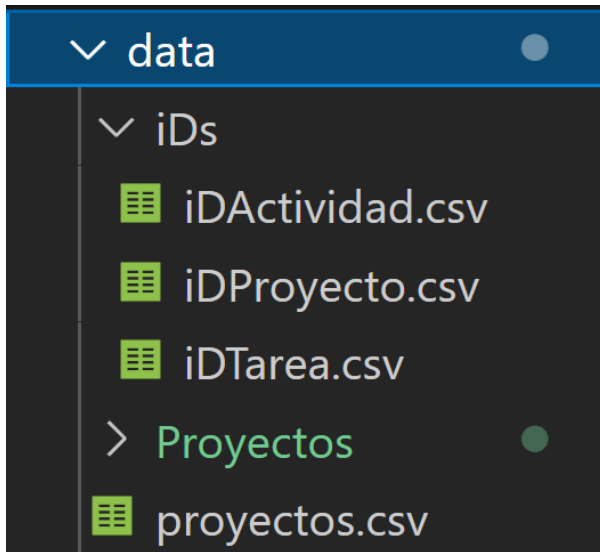
### Aplicación

- **Librería de un proyecto:** La aplicación posee una carpeta por proyecto creado en la cual se encuentran archivos para registrar su contenido y avance. Cada carpeta de proyecto posee un archivo de participantes, otro de actividades, otro de actividades, otro de reporte de usuarios y uno último de reporte del proyecto en general; los archivos se relacionan entre si mediante números identificadores.



*Carpeta para el proyecto “Startup Menup”*

- **Librería de la aplicación misma:** La aplicación guarda su información general en 4 archivos, tres de ellos guardan identificadores que se asignan a una clase respectivamente y en el restante se guarda información general de los proyectos creados.



*Sistema de archivos de la aplicación*

- **Clases**
  - **Un proyecto tiene:**
    - Un nombre
    - Una descripción
    - Una fecha inicial
    - Una fecha final
    - Un identificador
    - Un conjunto de tipos de actividades
    - Un conjunto de participantes
    - Un conjunto de tareas
  - **Una tarea tiene:**
    - Un identificador
    - Un nombre
    - Una descripción
    - Un tipo de tarea
    - Uno o varios autores
    - Una fecha inicial
    - Una fecha final
    - Una hora inicial
    - Una hora final
    - Un conjunto de actividades
    - Un conjunto de subtareas que poseen todo lo anterior
  - **Una actividad tiene:**

- Un título
  - Una descripción
  - Un tipo de actividad
  - Una fecha inicial
  - Una fecha final
  - Una hora inicial
  - Una hora final
  - Un tiempo de realización
  - Un autor
  - Un identificador
- **Un participante tiene:**
    - Un nombre
    - Un correo
    - Un identificador de propiedad del proyecto
    - Un conjunto de actividades

- **Consideraciones adicionales de diseño**

- Todas las clases se identifican por un id único menos la clase de Participante, la cual se identifica por el nombre.
- Para la WBS en principio se pensó usar Map y que pudiera haber unos dentro de otros pero no se tomó esta opción debido a que no permitía formar paquetes de actividades(tareas). Por lo anterior y además porque actividades y tareas poseían atributos y posibles métodos futuros a implementar para reportes, se decidió crear una clase para tarea que tuviera en sus atributos un ArrayList de tareas y otro de Actividades, de esta manera, podía haber paquetes de actividades y una organización jerárquica.
- Se decidió tener una carpeta por proyecto para no tener archivos muy grandes y con información dispersa.
- Los ids de actividades, proyectos y tareas hubieran podido estar en un mismo archivo csv pero no se tomó esta opción ya que en caso de querer agregar un id para algo más en el mismo archivo, la manera en la que se cargaban los datos debía cambiar.
- Se creó una clase Plataforma para manejar temas de estructura, como, por ejemplo, las rutas de los archivos y tener métodos que arrojaran las rutas de diferentes archivos en diferentes proyectos debido a la organización de carpeta por proyecto.

- **UML:** [https://drive.google.com/file/d/1Sg0pn\\_1-ftbPbfwE3Go3YqGzc1k2IQM1/view?usp=sharing](https://drive.google.com/file/d/1Sg0pn_1-ftbPbfwE3Go3YqGzc1k2IQM1/view?usp=sharing)

En esta versión del proyecto se busca agregar WBS (Work Breakdown Structure) como un ArrayList para poder organizar las actividades de cada proyecto por separado. Estos se van a implementar a partir de una nueva clase llamada “Tarea” incluido en el paquete “model”. Otro cambio implementado fue el del GUI para agregar elementos que faltaron en la anterior versión del proyecto, siendo los cambios más evidentes la creación de 2 nuevas clases de la interfaz:

La lógica de la aplicación no ha recibido cambios significativos y siguen operando con los mismos algoritmos y estructuras.

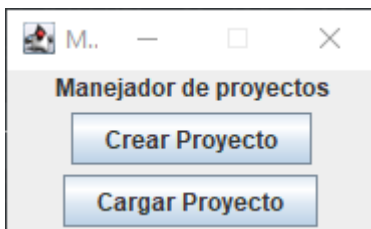
### **Manejo de errores:**

Se han implementado manejos de errores en la aplicación, las cuales la mayoría dependen de la interacción con el usuario para poder continuar o detener.

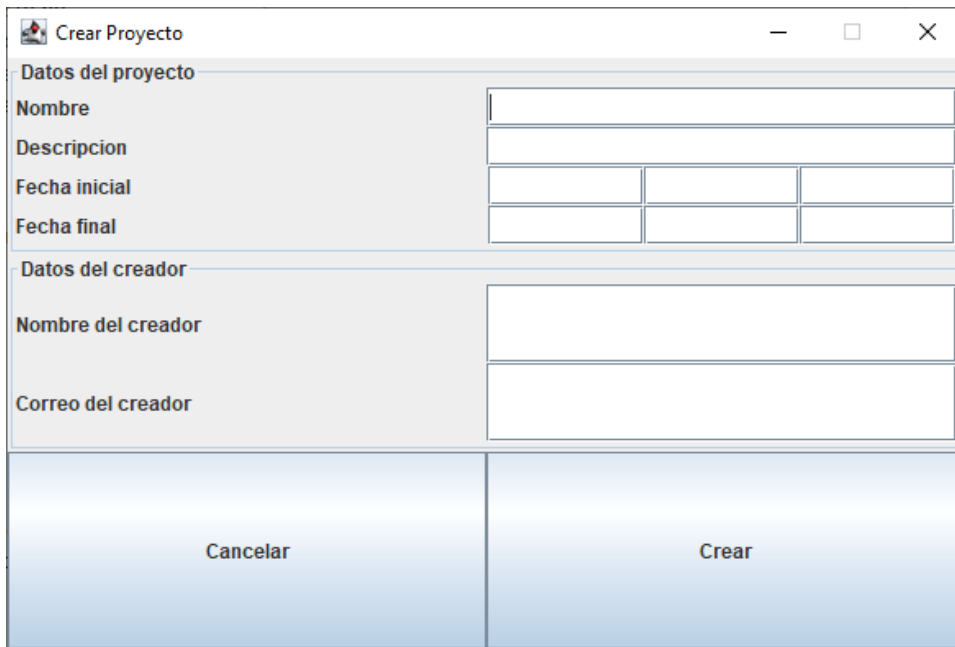
- Al momento de cargar un proyecto se le solicita al usuario que digite únicamente el nombre del proyecto que quiera cargar. Esta ventana le indicara al usuario si el nombre que introdujo no corresponde a algún proyecto o si ha dejado el JTextField vacío. Si cierra la ventana, se le informara que cancelo la carga del proyecto.
- Al momento de correr la aplicación, se le puede notificar al usuario si falta algún archivo importante para la aplicación, así como cualquier otro error posible. Sin embargo, esto no detiene a la aplicación de ejecutarse e intentar realizar las operaciones de cargar proyecto o crear proyecto sin tener éxito. Sin estos archivos, el programa podría crear errores graves que serán difíciles de manejar ya que puede asignar el mismo ID a otro proyecto o sobrescribir los datos de un proyecto al crear o guardar datos.

### **Interfaz Gráfica:**

La aplicación empieza con una ventana pequeña que ofrece 2 opciones (crear proyecto o cargar proyecto).

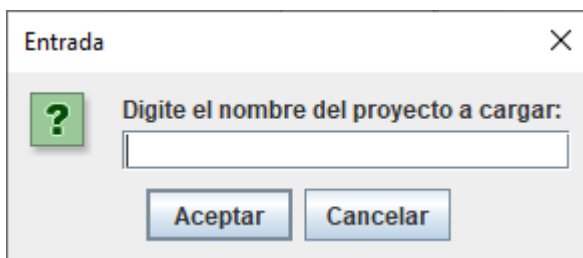


Si se selecciona la opción de crear proyecto, se abrirá una nueva ventana que consta de varios JTextField, 3 JPanel, varios JLabel y 2 JButtons. Estos les solicitan al usuario que les ingrese los datos del proyecto.



A Java Swing dialog box titled "Crear Proyecto" with standard window controls (minimize, maximize, close). It is divided into two main sections: "Datos del proyecto" and "Datos del creador". The "Datos del proyecto" section contains four labels: "Nombre", "Descripcion", "Fecha inicial", and "Fecha final". "Nombre" and "Descripcion" are followed by single-line text input fields. "Fecha inicial" and "Fecha final" are followed by three-column date pickers. The "Datos del creador" section contains two labels: "Nombre del creador" and "Correo del creador", each followed by a single-line text input field. At the bottom, there are two large, light-blue buttons: "Cancelar" on the left and "Crear" on the right.

Si se opta por cargar un proyecto, se le solicitara al usuario a través de un JOptionPane que ingrese únicamente el nombre del proyecto. Dependiendo del resultado, saldrá otro JOptionPane si cancelo la carga del proyecto, si no existe algún proyecto con ese nombre o si el usuario dejo una cadena de texto vacía



A small Java Swing dialog box titled "Entrada" with a close button. It features a green square icon with a white question mark on the left. To the right of the icon is the text "Digite el nombre del proyecto a cargar:". Below this text is a single-line text input field. At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Administrar proyecto

Administrar proyecto

Nombre: Startup Menupp

Descripcion: d de ventas de los sus vendedores

Fecha inicial: 28/08/2021

Fecha final:

ID: 0

Movimiento del proyecto

Movimiento del proyecto

Guardar

Opciones

Mostrar participantes

Mostrar tipos de actividad...

Mostrar actividades

Generar reporte

Sucesivamente cada boton en el programa que requiera de hacer un cambio, guardar o cargar datos puede mostrar dos `JOptionPane` mostrando si se pudo completar la operación o si hubo algun problema.

Todos los botones de la ventana de administrar datos del proyecto llevan a disntintas ventanas según sea lo que se quiera hacer, salvo el boton de Generar reporte, que mostrara dos `JOptionPane` dependiendo de si se pudo generar el reporte o si hubo un error.

El `JTextFrame` de Fecha final en este caso puede estar en blanco ya que no es necesario que un proyecto tenga una fecha final.

Hay una ventana que fue modificada:

- **VentanaRevisarTiposActividades:** Esta ventana no tenía una estructura visual esperada, ya que mostraba los botones del lado derecho en lugar de mostrarlos en la parte baja de la ventana. Adicionalmente, se ha agregado un `JList` que le mostrara todos los tipos de actividades que hayan sido registradas en el proyecto.

Como se mencionó previamente, se han añadido 2 nuevas clases que representan 3 nuevas ventanas. Estas son:

- VentanaCronometrarActividad: Se muestra cuando se va a crear una nueva actividad en tiempo real.
- VentanaAniadirParticipantes: Se muestra cuando se quiere agregar participantes a un proyecto ya existente. Consta de 2 JTextField y 1 JButton y se le notifica al usuario si se pudo crear un nuevo participante o no a través de JOptionPane