

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.

Análisis del dominio

- **Requerimientos funcionales:**

- **Nombre:** crear un proyecto. **Entradas:** ninguna. **Resultados:** una carpeta con el nombre del proyecto creado y el despliegue de un menú de opciones para empezar a trabajar con el nuevo proyecto.

- **Nombre:** cargar un proyecto. **Entradas:** El nombre del proyecto que se desea cargar. **Resultados:** el ambiente de la aplicación y sus variables se destinan a los atributos del proyecto y se despliega un menú de opciones para empezar a trabajar con el proyecto cargado.

- **Nombre:** agregar un nuevo participante a un proyecto. **Entradas:** información del nuevo participante como nombre y corre. **Resultados:** el nuevo participante y su información se registran en el archivo de participantes y se agregan a la lista de participantes del proyecto.

- **Nombre:** agregar una nueva actividad al proyecto. **Entradas:** atributos de la actividad como a qué tarea pertenece. **Resultados:** se agrega la actividad a la lista de actividades del proyecto, se registra en el archivo de actividades y se agrega a la WBS en la tarea a la que pertenece.

- **Nombre:** agregar una fecha final a un proyecto ya creado al proyecto. **Entradas:** la fecha. **Resultados:** se actualiza el atributo de fecha final del proyecto a la fecha ingresada.

- **Nombre:** agregar un tipo de actividad al proyecto **Entradas:** el tipo de actividad. **Resultados:** se agrega el tipo de actividad a la lista de tipos de actividades del proyecto.

• **Nombre:** guardar un reporte de los participantes del proyecto. **Entradas:** ninguna. **Resultados:** se guarda un archivo en la carpeta del proyecto con un reporte de cada participante del proyecto con datos como el tiempo total invertido en el proyecto.

• **Nombre:** guardar un reporte del avance del proyecto. **Entradas:** ninguna. **Resultados:** se guarda un archivo en la carpeta del proyecto con un reporte del avance del proyecto en comparación a lo planeado.

• **Nombre:** agregar una nueva tarea al proyecto. **Entradas:** nombre de la nueva tarea. **Resultados:** se agrega la tarea a la WBS y se agrega en el archivo de tareas del proyecto.

• **Nombre:** guardar el proyecto y salir al menú principal. **Entradas:** ninguna. **Resultados:** se guardan todos los cambios hechos y se muestra el menú principal de la aplicación.

• **Nombre:** salir de la aplicación. **Entradas:** Ninguna. **Resultados:** se cierra el menú principal de la aplicación y se termina la ejecución de esta misma.

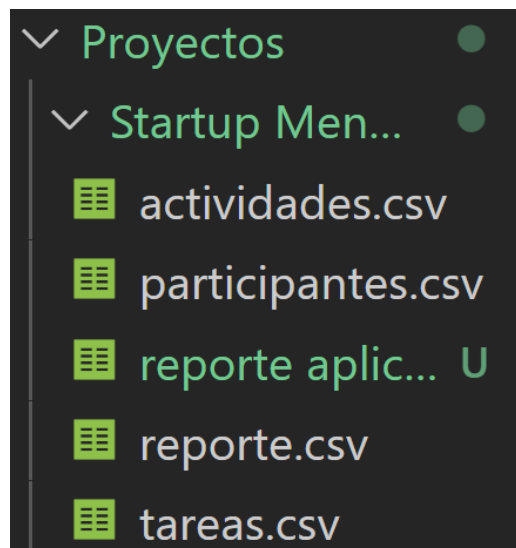
• **Restricciones:**

- Las tareas y actividades del proyecto están organizadas con un sistema de WBS.
 - Las tareas contienen actividades y no viceversa.
 - La WBS tiene una organización jerárquica.
 - Los tipos de actividades y tareas los define el usuario y pueden ser diferentes entre proyectos.
 - No hay límite de tareas ni de actividades dentro de una tarea.
 - Al registrar una actividad se debe indicar a qué tarea está asociada, de lo contrario, no se podrá agregar. Si la tarea no existe, se debe crear primero.
 - El tiempo que tomó realizar una tarea es a sumatoria del tiempo que tomó realizar todas las actividades que están relacionadas a la tarea.
 - La fecha de finalización de una tarea es la fecha de finalización de la actividad que la completa.

- Al momento de crear un proyecto, tarea, actividad, participante o cualquier otro componente, todos los campos requeridos son obligatorios menos la fecha de finalización en la creación de un proyecto, la cual puede establecerse o modificarse cuando se quiera.
- Cuando se crea un proyecto, se debe crear a la vez un usuario, que será el que se registrará como dueño del proyecto.
- Solo puede haber un dueño en cada proyecto.
- No se puede agregar una actividad o tarea con cierto tipo de actividad si el tipo de actividad no existe, debe crearse primero.
- Todos los componentes de la aplicación se identifican por números únicos, menos los participantes, por ende, no puede haber participantes con el mismo nombre.
- No es posible pasar al menú de opciones para un proyecto si no se ha cargado ningún proyecto.

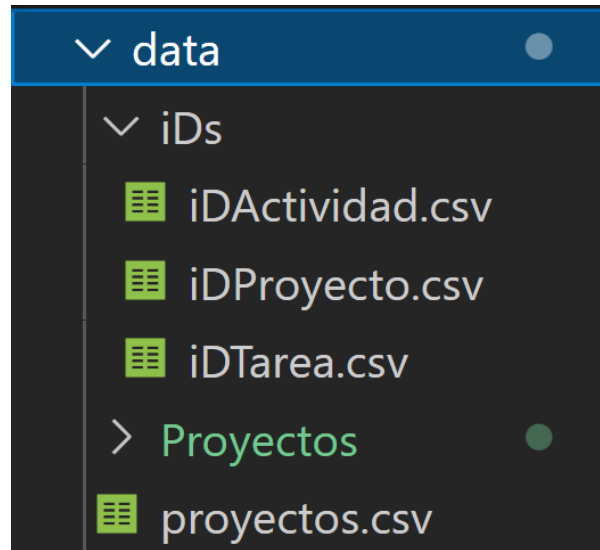
Diseño de la aplicación

- **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.

- Se trabajó un patrón MVC.

- **UML:**

https://drive.google.com/file/d/1Sg0pn_1ftbPbfwE3Go3YqGzc1k2IQM1/view?usp=sharing

Diseño de la interfaz



Ventana principal

CREAR PROYECTO

Nombre

TextField

Descripción

TextField

Fecha inicial

Three date input fields for the start date.

Fecha final

Three date input fields for the end date.

Información del dueño

Nombre

TextField

Correo

TextField

Crear
(JButton)

Ventana para crear un proyecto

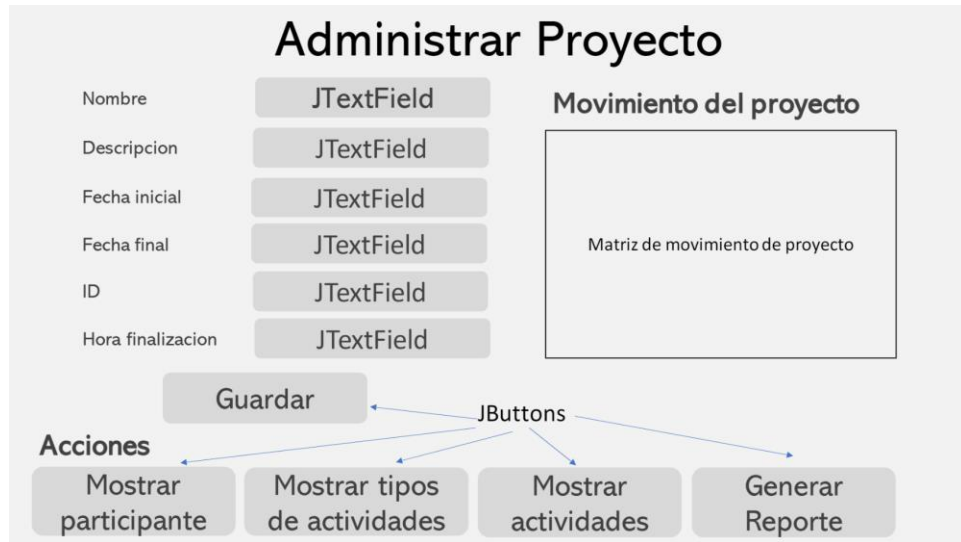
CARGAR PROYECTO

Nombre del
proyecto

TextField

Cargar
(JButton)

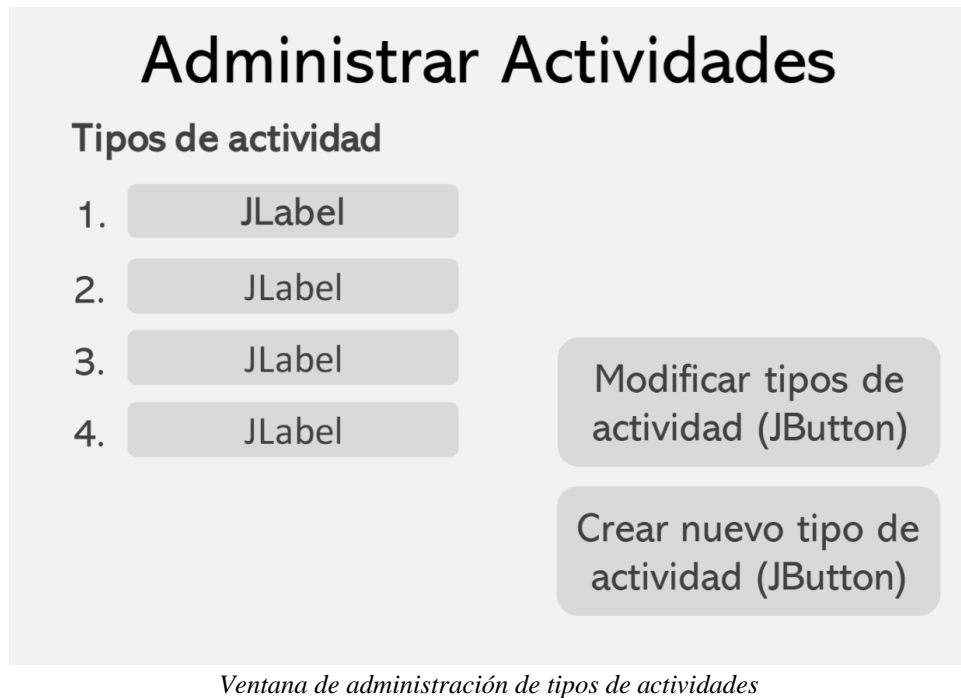
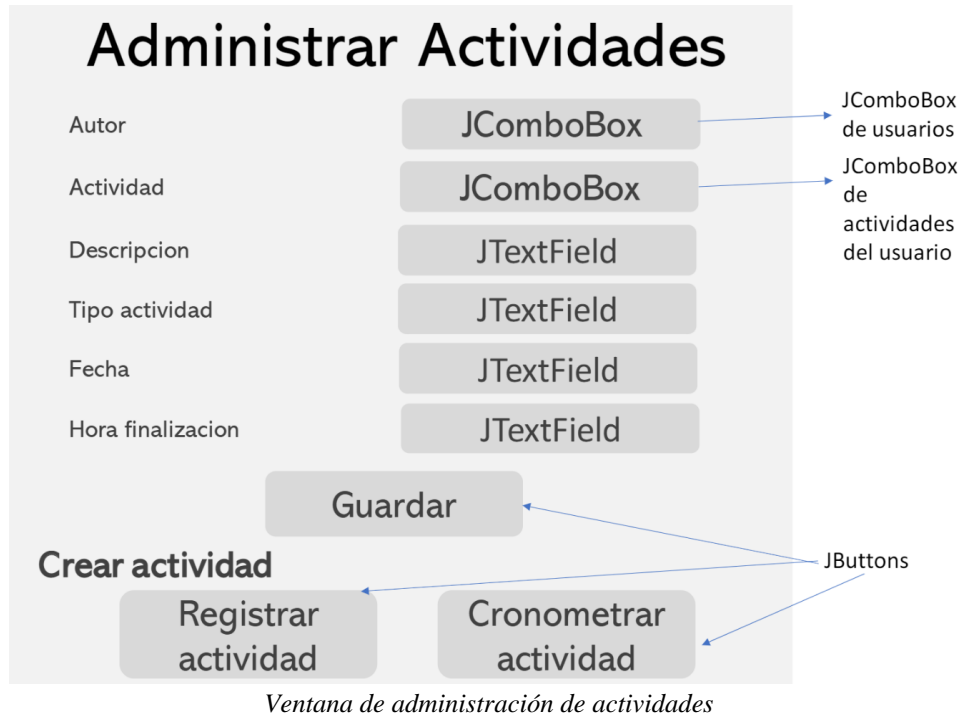
Ventana para cargar un proyecto ya creado



Ventana principal de un proyecto cargado



Ventana de administración de participantes



Crear nuevo tipo de actividad

Nombre del
nuevo tipo

TextField

Crear
(Button)

Ventana de creación de tipo de actividad

Administrar Actividades

Crear actividad en curso

Título

TextField

Descripción

TextField

Tipo

TextField

Autor

TextField

Tiempo
cronometrado

Label

Iniciar/Pausar
cronómetro
(Button)

Finalizar
cronómetro
(Button)

Ventana de creación de actividad en tiempo real

Administrar Actividades

Registrar actividad finalizada

Título	TextField
Descripción	TextField
Tipo	TextField
Autor	TextField
Fecha de inicio	TextField
Fecha de fin	TextField

Registrar
(JButton)

Ventana de creación de actividad sin tiempo real

Diseño de 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.
- En caso de que no se encuentren las rutas de los archivos por cualquier momento, existe una excepción que arroja una ventana informando el error pero que permite continuar con la ejecución del programa.
- Cada restricción descrita en este documento posee un manejo de error en el código, por ejemplo, si el usuario está agregando una actividad con un tipo de actividad que no existe, se le informara este evento por medio de una ventana.

