

Entrega 1 Proyecto 3

David Burgos - 201818326

Wyo Hann Chu Méndez - 202015066

Juan Daniel Sepúlveda - 202113067

Aplicación

En esta entrega no cambiamos de gran forma el modelo. Inicialmente adaptamos la clase aplicación para que retornara el texto como parámetro de diferentes formas, esto facilitará el mostrar información desde la interfaz gráfica. Los cambios más sustanciales en el modelo ocurrieron en cuanto a la toma del tiempo, se agregaron nuevas funcionalidades que envían el tiempo total y el tiempo promedio por actividad directamente a la GUI.

Además, para esta nueva entrega de vamos a mantener básicamente nuestro mismo diseño de clases las cuales solo tienen implementaciones extras para los nuevos requerimientos, el cambio más relevante es la creación de la clase paquete, la cual se crea para tener una nueva implementación para guardar la información y datos de la aplicación y tener persistencia, solo se piensa requerir esta clase debido al bajo acoplamiento del diseño. Con esto nos ayudamos para implementar los nuevos requerimientos funcionales: Avance del proyecto con respecto a el mismo, sus estadísticas y paquetes independientes, Calidad de la planeación con respecto a su asertividad en tiempos propuestos y los errores de estimación, datos e información del equipo y resumen del proyecto incluyendo los tiempos totales, la cantidad de tareas pendientes, el avance, etc.

Las clases que mantienen la misma idea:

- **Aplicación (Structurer):** es la encargada de manejar los proyectos que los usuarios hayan creado con anterioridad y de crear nuevos usuarios y proyectos. Además, debe ser capaz de hacer las siguientes cosas: o Llevar la cuenta del tiempo que le toma a un usuario el realizar cierta actividad, en otras palabras, contar con un cronómetro. Este debe poder pararse y reanudarse cuando el usuario lo desee. o Mostrar un reporte de las actividades que cada participante haya realizado en un proyecto. Este debe contar con información como: tiempo total invertido, tiempo promedio por tipo de actividad, tiempo por cada día, etc.
- **Usuario (Info Holder):** es una clase que puede ser utilizada por una persona, ya sea participante o no de un proyecto, el cual tiene la posibilidad de registrar participaciones en un proyecto y actividad, también debe poder modificar aspectos como la duración de la actividad o la fecha de creación del proyecto. Además, el usuario debe tener la potestad de registrar una actividad hecha por algún otro participante.

- Actividad (Info Holder): es la encargada de contener la información de una actividad que haga parte de un proyecto. Cada una de estas debe tener un título, una descripción, un tipo (“Documentación”, “Implementación”, “Pruebas”, “Investigación”, “Diseño”, etc.), fecha de realización, hora de inicio, hora de finalización y el participante que la realizó. También se deben poder modificar las horas en caso de que se haya trabajado mientras no se haya tenido acceso al sistema.
- Persistencia (Interfacer): es la clase encargada de guardar la información que se haya obtenido mientras el programa haya estado activo. Lo anterior, con tal de que la información quede guardada en algún lugar en caso de que se cierre la aplicación.
- Participante (Info Holder): Es la clase encargada de crear los usuarios o participantes que hacen parte de un proyecto y de una o varias actividades. Este puede registrar que realizó una actividad u otro usuario puede hacerlo por él.

Interfaz gráfica

Estructura de la interfaz gráfica:

Toda la aplicación inicia en la ventana principal, en esta se hace uso del BorderLayout para poder acomodar varios paneles que contengan toda la funcionalidad requerida de la aplicación, además, se inicializa el modelo para empezar a hacer uso de aplicación. Desde esta se inicializa el resto del modelo tal cual como se hizo anteriormente en el proyecto 1, se inicializa la impresora, misma que es encargada de que la información obtenida en otras iteraciones de la aplicación pueda volver a ser utilizada. Una vez inicializada esta clase, cuando el usuario hace uso de la interfaz gráfica se llaman a las clases del modelo que se encargan de manejar los proyectos, actividades y usuarios.

Por otra parte, en la clase Ventana principal se crean 3 paneles: el primero se encuentra ubicado en el norte del BorderLayout, aquí se guardan los datos que se le piden al usuario una vez se inicializa la aplicación (usuario y correo). El panel que se ubica en la posición central del BorderLayout es el encargado de cambiar, dependiendo de la funcionalidad que el usuario requiera, una vez el usuario haga uso de los botones que se encuentran ubicados en el sur del BorderLayout estos botones se encuentran organizados dentro de un CardLayout que maneja el panel ubicado en la posición anteriormente dicha, dentro de este se guardan los botones encargados de cambiar la funcionalidad de la aplicación según lo necesite el usuario (“Crear nuevo proyecto”, “Crear nueva actividad”, etc.).

En PanelCrearProyecto se encuentra el código destinado a aparecer una vez el usuario haga click en el botón “Crear nuevo proyecto” ubicado en la parte de debajo de la aplicación. Aquí se cambia el layout del panel ubicado en el centro del BorderLayout para que cree un GridLayout donde se ubican los JLabel y JTextField de manera contigua, mismos con los que el usuario podrá interactuar para crear un nuevo proyecto. En la parte de abajo a la izquierda en el GridLayout se encuentra un botón titulado como “Guardar nuevo proyecto”, una vez el usuario de click en este botón todos los JTextField del panel se volverán a poner en blanco y

la información dada por el usuario será guardada en un archivo .txt que la impresora se encargará de leer.

En PanelCrearActividad pasa algo parecido que en PanelCrearProyecto, la diferencia más sustancial está dada en la cantidad de filas que el GridLayout contiene, ya que, se necesitan más JLabels y JTextFields para llenar toda la información requerida para la creación de una actividad.

La clase de VerProyecto es la encargada de mostrarle al usuario desde la interfaz gráfica el proyecto en el que se encuentra trabajando actualmente y funciona también como el panel que primero se le muestra al usuario. Por defecto, cuando el usuario digita su usuario y correo se crea un proyecto vacío, si el usuario quiere crear un nuevo proyecto, o, en otras palabras, cambiar de proyecto necesitará hacer uso de los botones de “Crear proyecto” o “Cambiar de Proyecto”. Y las nuevas implementaciones son solamente funcionalidades que se le agregan a los botones creados en la misma clase.

Manejo de errores

Analizando el programa vamos a catalogar los errores en dos grandes categorías, los que podemos manejar con elementos propios de la aplicación y los que necesitan de la interacción del usuario para solucionarse. La primera categoría son las situaciones que se pueden evitar o solucionar desde el propio código, un ejemplo son los campos de texto que necesariamente son un número o una cadena de caracteres, si ajustamos los parámetros de los cuadros de texto podemos evitar estas situaciones y no necesitamos escalar a otra clase extra.

Por otra parte, podemos analizar las situaciones que necesitan la interacción de un usuario, estas situaciones usualmente surgen porque se intenta crear un objeto que necesita un componente no inicializado. Por ejemplo, no es posible que se añada una actividad o tarea sin crear antes un proyecto, en estas situaciones que incluyen diferentes clases es muy útil el manejo de excepciones, debido a que, solo se envía una excepción que notifique al usuario de su error y retornamos al estado de ejecución normal.

Por último, planeamos un set de pruebas para la aplicación que revisen las funcionalidades de guardar datos y los reportes generados. El primer set busca que la función GenerarReporte() que tiene proyecto, paquete y actividad esté funcionando correctamente, en estos casos se crea la clase que vamos a probar con unos valores que ya conocemos y comparamos el resultado de la función con no hecho a mano. Para probar las funciones que guardan datos solo vamos a darles casos de prueba y revisar los archivos generados.