

Diseño y Programación Orientada a Objetos – Sección 6

Jerónimo Vargas Rendon - 202113305

Nicolas Camargo Prieto – 202020782

Documento de Diseño

ANALISIS

Descripción de Requerimientos Funcionales:

HISTORIAS DE USUARIO

1. Como participante de un proyecto, quiero poder tener un sistema que registre todo el proceso de desarrollo de dicho proyecto, con el fin de conocer los miembros que participaron e información de las diferentes tareas que se efectuaron los miembros del grupo.

- Registrar en un sistema información del proyecto.
- Ver los miembros del proyecto.
- Ver registro de actividades.
- Ver información calculada de los datos.

2. Como dueño del proyecto creado, quiero poder registrar nuevos participantes ya que inicialmente soy el único, mientras que es muy probable que un proyecto se conforme de varios participantes que aportan al desarrollo de este mismo.

- Registrar los miembros del proyecto (Crear un participante).
- Registrar nombre y correo del participante.

3. Como participante del proyecto, deseo poder registrar en el sistema las actividades que voy realizando en el proyecto, para dejar soporte de la tarea que realicé, con la fecha en la que se realizó, información que la caracteriza y tiempos en el cual se desarrolló.

4. Como participante del proyecto, quiero poder hacer varios registros o commits en una misma tarea, debido a que la actividad a realizar puede que tarde días en realizarse.

5. Como participante del proyecto, quiero poder cronometrar el tiempo que tardo en realizar una actividad del proyecto, además deseo poder detener el cronómetro en caso de que deba interrumpir el desarrollo de la actividad.

6. Como dueño del proyecto, quiero poder recibir un reporte que informa de las actividades realizadas por cada miembro y diferentes características calculadas.

- Ver total actividades.
- Ver Tiempo total invertido.
- Ver Tiempo promedio por actividad.

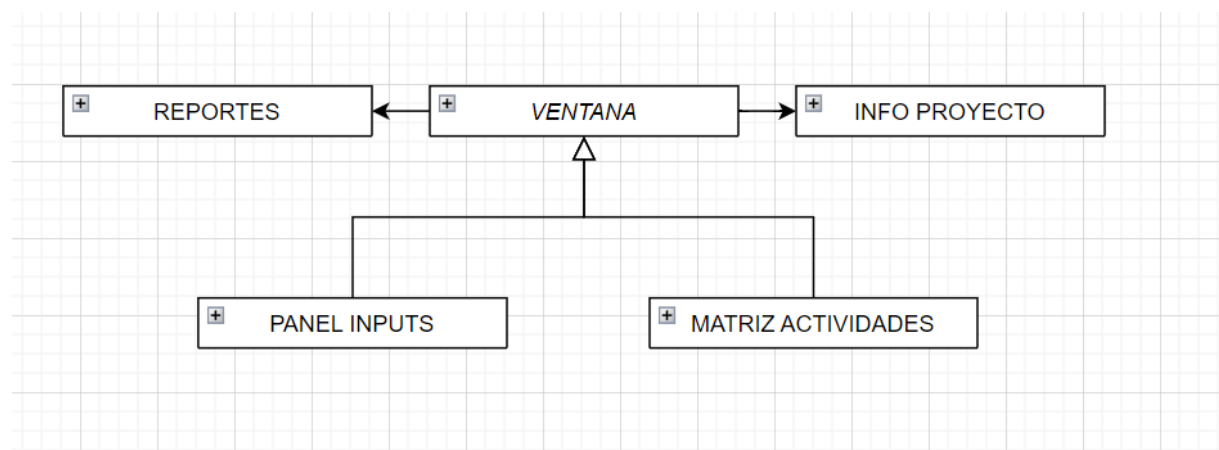
Restricciones:

Aunque el diseño del proyecto incluye la mayoría de los aspectos esenciales y requerimientos funcionales, también este cuenta con unas cuentas restricciones. Por un lado, el proyecto solo puede ser realizado por dos personas, quienes deben tener en cuenta los cambios realizados por el otro. Se debe usar eclipse (Java) para su realización, y de la misma forma, todos los cambios deben ser subidos a GitHub. Por otro lado, la modificación de datos en el registro puede generar conflictos con la fecha y hora en la que se realizaron las actividades. A sí mismo, establecer en el diseño varias tareas con el mismo nombre, puede suponer conflictos a la hora de guardar la información. Por último, el tipo de actividad que debe ser definida (Documentación, Implementación, Pruebas, etc.) deben ser seleccionadas previamente, pues hacen parte de un arraylist dentro de la clase proyecto, que es importante para interactuar con el resto de las clases.

Diseño Interfaz Grafica

En relación a los requerimientos funcionales anteriores se inició un diseño desarrollado en SWING con el fin de generar una interacción con el usuario más amena, dinámica e intuitiva.

AGRUPACION DE ELEMENTOS DE LA INTERFAZ GRAFICA



ITERACIONES PRINCIPALES

Para comenzar a desarrollar la interfaz de identificaron los elementos necesarios para crear una buena interacción con el usuario, es por esto que se ha decidido trabajar sobre 5 elementos.

La ventana la cual va a ser el JFRAME principal ya que es la base de la aplicación, de la mano con la venta se desarrollarán dos paneles los cuales estarán constantemente ofreciéndole un requerimiento al usuario. Uno de los paneles es el encargado de todos los inputs de la aplicación para poder recolectar todos los datos que el usuario desee suministrarle al software de Proyectos, y mientras tanto otro panel está funcionando dinámicamente para ir mostrando una matriz con todas las actividades desarrolladas por los participantes.

PANEL INPUTS

Proyecto 2

Crear Proyecto

Nombre del proyecto:

Descripcion del proyecto:

Fecha Inicio (yyyy-mm-dd):

Fecha Final (yyyy-m...

Crear

Modificar fecha final del Proyecto

Fecha Final (yyyy-m...

Modificar

Agregar Participante

Nombre del participante:

Correo del participante:

Agregar

Eliminar Participante

Nombre del participante:

Eliminar

Mostrar Informacion del proyecto

Agregar Actividad

Nombre del autor de la actividad:

Titulo de la activi..

Descripci...

Tipo de Activid...

Fecha (yyyy-mm...

Este Panel Sufrió diversas iteraciones durante el desarrollo de código, ya que, al chequear los requerimientos del proyecto, diversas veces encontramos errores en los inputs y además que algunos hacían falta para cumplir por todo lo establecido en el modelo de negocio.

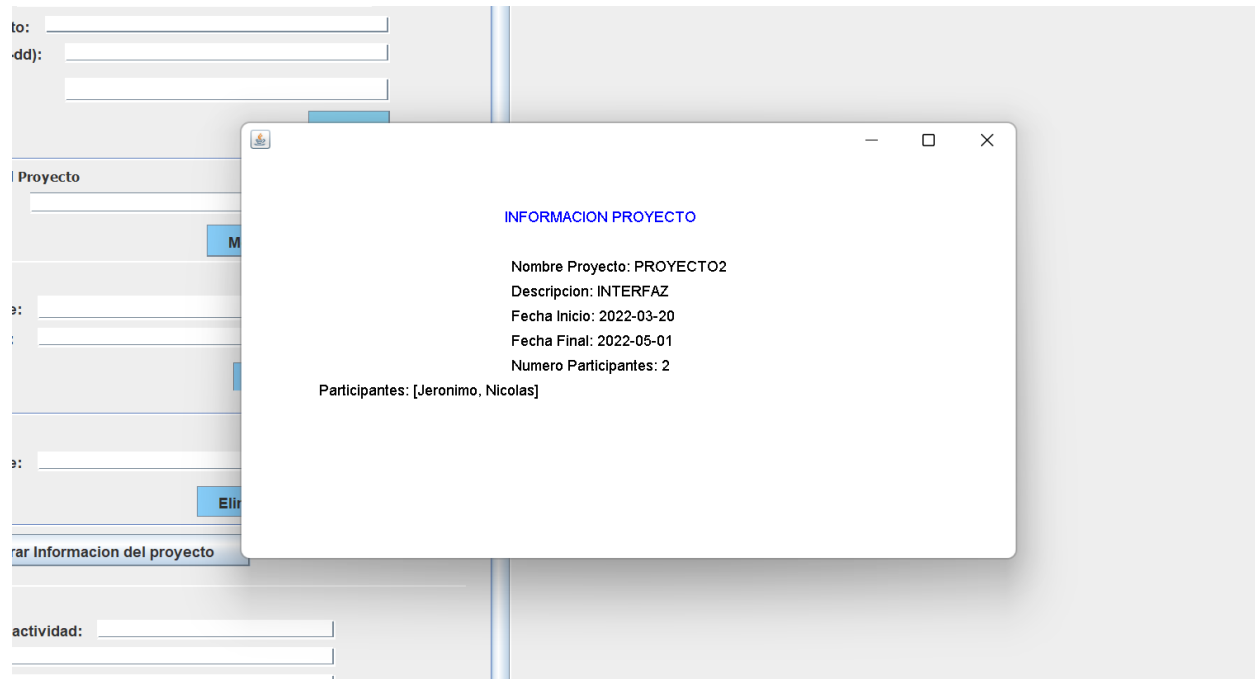
PANEL MATRIZ ACTIVIDADES

MATRIZ ACTIVIDADES			
			Nº: 4
Título: Observador	Autor: Jeronimo	Tipo: Implementacion	2022-05-02
Título: persistencia	Autor: Jeronimo	Tipo: Implementacion	2022-04-18
Título: Interfaz Inputs	Autor: Nicolas	Tipo: Implementacion	2022-04-28
Título: Bosquejo	Autor: Nicolas	Tipo: Diseño	2022-04-19

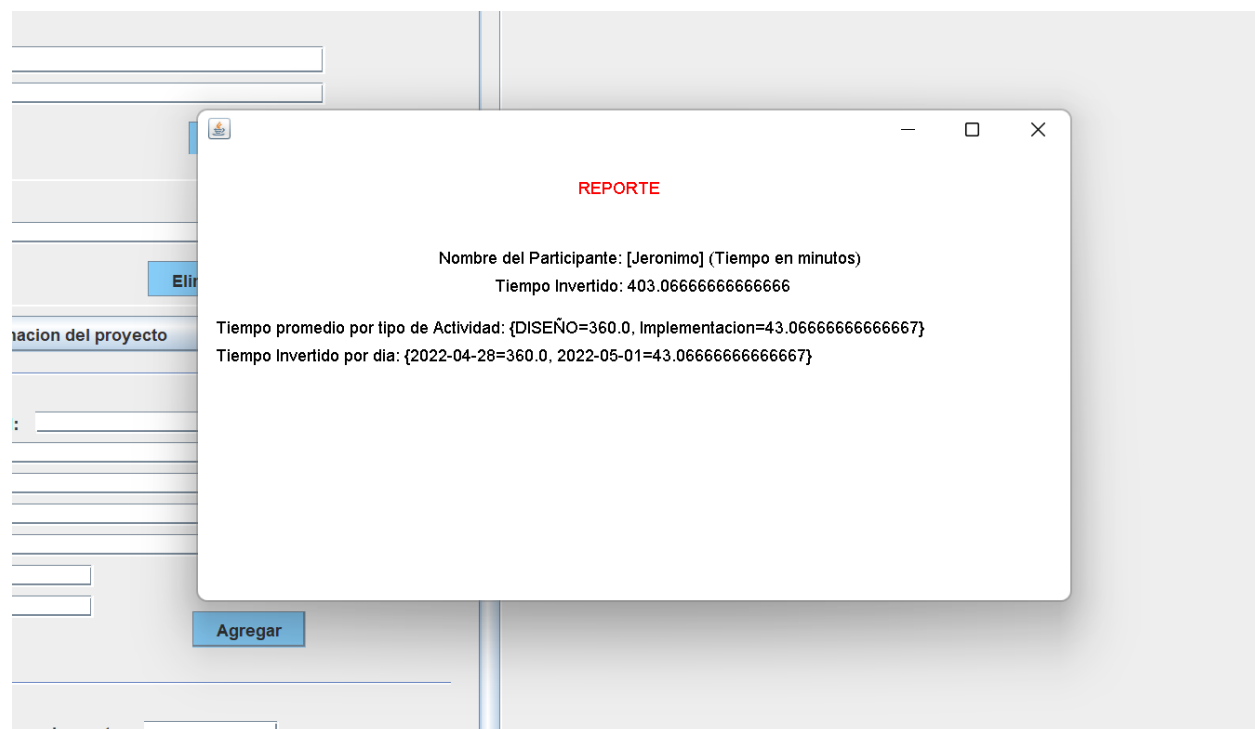
Mas allá de trabajar con un único JFrame, se decidió utilizar dos JFrame dinámicos que se inicializan en el momento que lo desea el usuario, por ejemplo, para mostrar información del proyecto y para realizar un reporte de algún participante este se inicializará en el momento

que lo desee el usuario. Esto se decidió de esta manera ya que este tipo de información es muy particular y debe ser mostrada en pantalla al instante que lo indique el usuario.

FRAME INFORMACIÓN PROYECTO

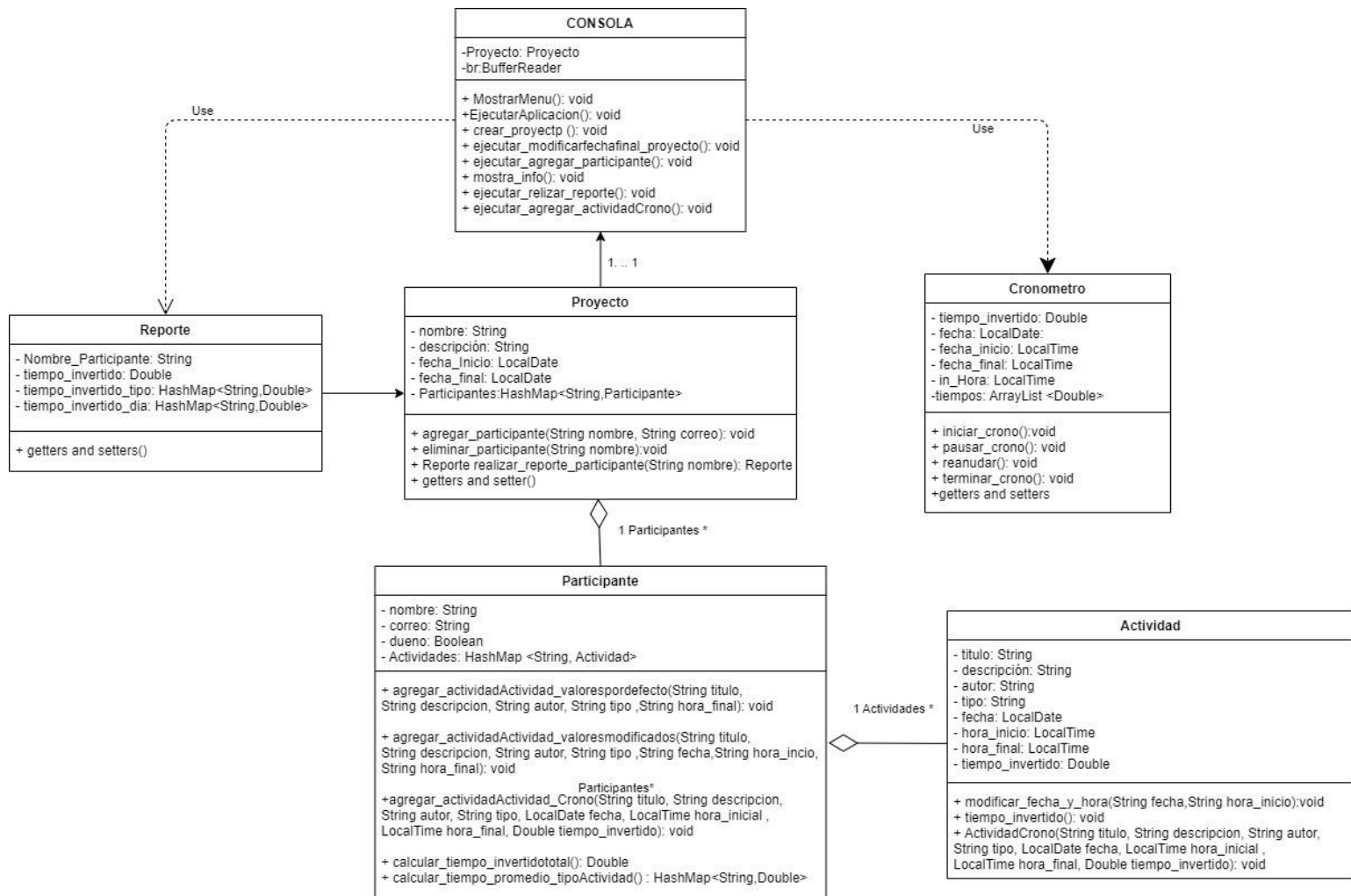


FRAME REPORTE PARTICIPANTE

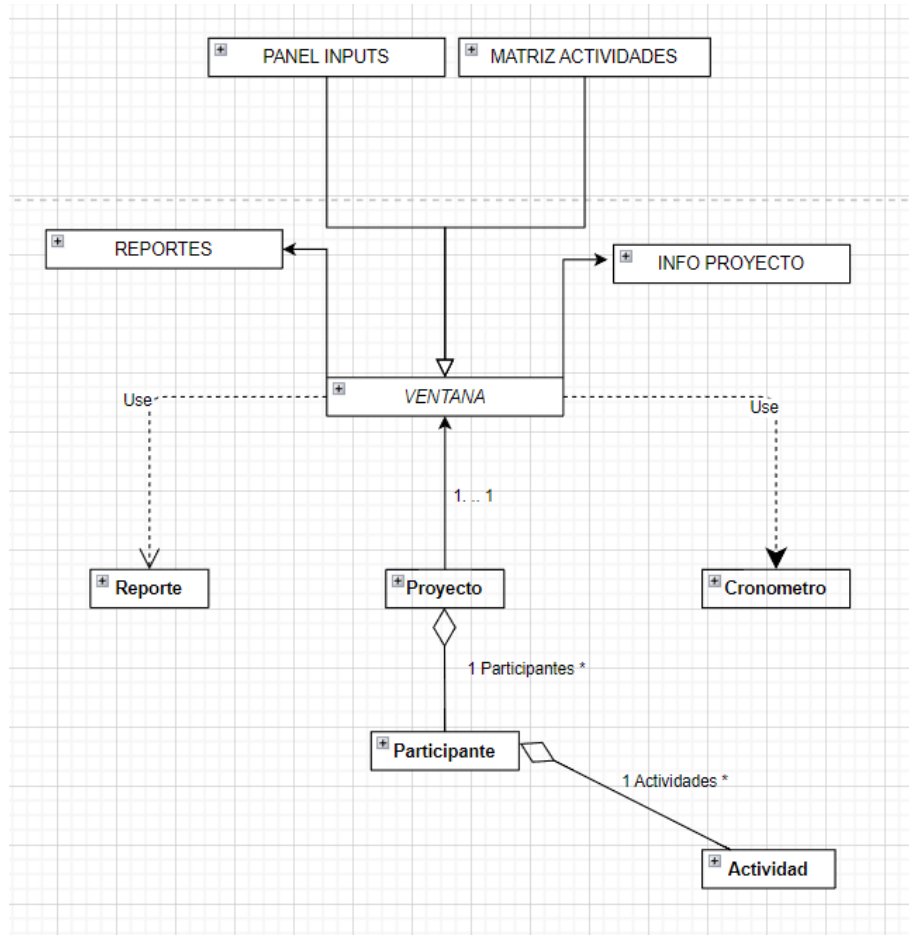


Estos dos frame al dar una información muy puntual y no cambiante se decidió hacerla en modo de un nuevo Frame para que tuvieran un cierto carácter de notificación para hacer más amena la interacción con el usuario.

Diagrama UML #4 Iteración (FINAL) Entrega 1

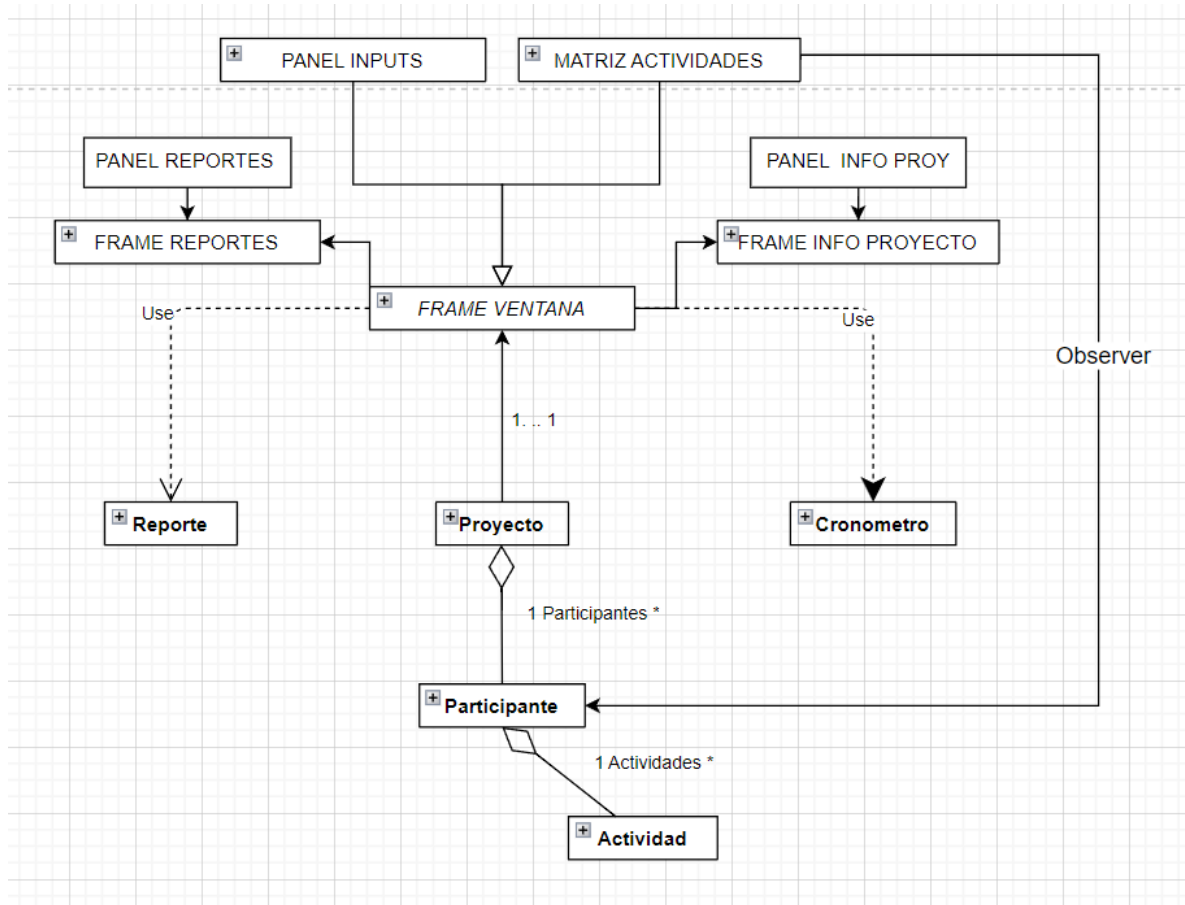


Si tenemos presente este diagrama vemos como la lógica de negocio se relacionaba con la consola primitiva desarrollada y para esta iteración donde se desarrolla un interfaz en SWING vemos que se consiguió plantear el siguiente diagrama UML.



Este diagrama permite ver que detrás de la interfaz también hay un desarrollo fuerte de código que optimizara la interacción con el usuario en relación a como lo era antes con la consola primitiva desarrollada en eclipse.

ITERACION FINAL (UML)



Justificación

En esta iteración vemos como se adopta el patrón de diseño MVC, es cual es bastante útil en el momento de desarrollar la matriz de actividades la cual es dinámica y debe ir actualizándose a medida que el usuario ingrese ciertos datos. En esta iteración vemos que la matriz de actividades implementa la interfaz de observador el cual le permite recibir una notificación de algún método de la lógica, y esta notificación se decidió hacer a través del objeto participante el cual es quien agrega las actividades. El objeto participante al ser observable se le permite notificar un objeto al observador y este objeto vienen siendo las diferentes actividades que se agregan para que a su vez se actualicen constantemente en la matriz de actividades.

Reflexión (Ventajas, Desventajas, Trade-offs):

Finalmente, se evidencia que el proceso de diseño que se llevó a cabo abarca de manera eficiente lo que se desarrolla en el programa, pues de manera significativa se denota la separación entre la lógica del programa y los nuevos aspectos implementados en la interfaz de la misma. De tal manera, resulta muy beneficioso el modelo planteado, ya que, la lógica del programa no afecta de ninguna forma importante a la interfaz, es decir, al despliegue de la información al usuario, resultando en procesos más simples que en definitiva son una ventaja tanto para el usuario como para posteriores desarrollos que se quieran implementar.

Sin embargo, la principal desventaja que se puede apreciar resulta en la gran extensión del frame ventana, pues este cuenta con más componentes de la interfaz que hacen que la complejidad de la vista sea más alta.

Por otro lado, en cuanto a los trade-offs se puede apreciar un claro cambio importante del modelo de despliegue de información e inputs por consola al nuevo modelo implementado de interfaz gráfica, que resulta mucho más cómodo para el usuario. De la misma forma, la implementación de la interfaz gráfica en más de una clase resulta más eficiente para la segmentación de la información y reducción de la complejidad del programa.