Nombre: Nicolas Perez Teran 202116903

Documento de Diseño

Iteración 1:

Inicialmente, identificamos un caso base para iniciar nuestra fase de diseño.

*Un joven llamado Carlos tiene la idea de un juego innovador para crear, razón por la cual contacta a sus cinco amigos para que lo ayuden; además, lo hace con la esperanza de que ellos conozcan a mas personas que se puedan unir al grupo. La idea de Carlos es ir conociendo a los integrantes y a los que mejor se desempeñen o mas interés tengan por ayudar les permitirá volverse un líder como el, para que contribuyan mas a las decisiones que se tomaran para el desarrollo. Por otro lado, Carlos necesita poder distribuir entre los integrantes las partes que se necesitan implementar al juego para facilitar la organización de como se hará todo.*

Con lo anterior podemos deducir que se necesitan los siguientes roles:

* Proyecto
* Líder
* Integrante
* Actividad

Y también se necesitaran cumplir las siguientes responsabilidades:

* Iniciar un proyecto nuevo.
* Agregar y asignar una actividad.
* Agregar integrantes.
* Definir cual integrante puede ser líder.

Con lo anterior, podemos identificar que se necesita especificar las responsabilidades a cumplir por nuestra solución.

Requerimientos Funcionales:

- Crear un nuevo proyecto: La idea es que aquí se especifiquen aspectos básicos del proyecto en general, como por ejemplo, el nombre, el líder, integrantes, etc.

- Añadir integrantes o lideres: Se necesita que el tamaño de integrantes y lideres sea variable, pues estos proyectos pueden requerir mucha colaboración.

- Mantener un seguimiento del proyecto: Se requiere poder manejar de manera mas simple todo lo que gira en torno al proyecto: avances dia por dia, tiempo estimado y transcurrido, y todo las actividades que se necesiten realizar.

Requerimientos no funcionales:

* Una interfaz exclusiva según el rol del participante: Se necesita que haya una interfaz que varíe dependiendo de el tipo de usuario que esta en uso.

Reglas:

* El proyecto se crea mediante la aplicación y debe ser el líder inicial quien maneja esto.
* El lider debe manejar todo el proyecto, mientras que los integrantes están limitados a cumplir con actividades.
* Los integrantes deben saber todo lo que les corresponde hacer a ellos.

Iteración 2:

Para esta parte, logramos identificar mas o menos las características de los roles:

* Aplicación: Director. Recordemos que este es quien maneja todas las relaciones y cambios de información entre los objetos del proyecto.
* Líder: Coordinador. Este puede mandar ordenes y hacer cambios en todo lo que tenga que ver con el proyecto que se maneja.
* Proyecto: Data holder. Esto es porque solo se limita a guardar y pasar información según lo pida la aplicación.
* Integrante: Data holder. Este objeto guarda la información de un integrante y la pasa o guarda según se requiera.
* Actividad: Data holder. Este objeto guarda todo lo que puede hacer una actividad.

Y con esto hecho, podemos actualizar y descomponer mas las responsabilidades.

* Iniciar un proyecto nuevo.
* Contar el tiempo transcurrido desde el inicio del proyecto.
* Mostrar las opciones disponibles para cada usuario.
* Agregar actividades para ser realizadas.
* Marcar las actividades que ya fueron realizadas.
* Definir las actividades que faltan por completarse.
* Mostrar cuantas horas se ha trabajado en una actividad.
* Mostrar cual integrante ha trabajado mas en el proyecto.
* Finalizar un proyecto.
* Agregar integrantes.
* Asignar el rol que ocupara un integrante (Líder o integrante normal).

Requerimientos Funcionales:

- Crear un nuevo proyecto: La idea es que aquí se especifiquen aspectos básicos del proyecto en general, como por ejemplo, el nombre, el líder, integrantes, etc.

- Añadir integrantes o lideres: Se necesita que el tamaño de integrantes y lideres sea variable, pues estos proyectos pueden requerir mucha colaboración. Además, se necesita poder cambiar entre un integrante o un líder.

- Mantener un seguimiento del proyecto: Se requiere poder manejar de manera mas simple todo lo que gira en torno al proyecto: avances dia por dia, tiempo estimado y transcurrido, y todas las actividades que se necesiten realizar.

- Se requiere poder crear y asignar actividades a un integrante o líder dado: Ya que para resolver un proyecto se necesita ir paso por paso, es necesario que cada integrante tenga claro que papel debe desempeñar.

Colaboraciones:

Dado que ya tenemos claro que papel desempeñara cada objeto, podemos hacer unas estimaciones sobre las relaciones que deben haber entre cada elemento.

* Aplicación: Posee relación con todos los objetos porque es aquel que se encarga de controlar el funcionamiento, comportamiento y actualización de cada uno de los elementos.
* Proyecto:
  + Tiene una relación con Aplicación: Aplicación será quien actualice toda la información en el proyecto y asignara las características de este.
  + Tiene una relación con líder: El proyecto necesita ser coordinado, y es el líder aquel que tiene control sobre todo lo que tenga que ver con el proyecto.
  + Tiene una relación con los integrantes: El proyecto es aquel que define cada cosa que debe ser realizado por los integrantes y les provee dicha información.
  + Tiene relación con Actividad: Para que el proyecto se encuentre finalizado, se requiere que se hayan cumplido una cierta cantidad de actividades y es por esto que el proyecto esta compuesto de una serie de actividades que el va creando.
* Líder:
  + Tiene una relación con proyecto: El líder se encarga de gestionar todos los aspectos del proyecto.
  + Tiene una relación con los integrantes: Se encarga de asignar que actividades deben cumplirse y puede asignarla a un integrante especifico. Además, tiene la habilidad de cambiar a un integrante a líder.
  + Tiene relación con las Actividades: El es quien puede decidir como van a ser las nuevas actividades. -El las define, pero no las crea-
* Integrante: Aparte de las anteriores ya mencionadas, tiene una relación con las actividades; esto porque el integrante es aquel que las completa y puede dar una actividad por realizada.
* Actividad: Todas sus relaciones ya fueron especificadas.

Reglas:

* La aplicación debe ser quien regule todas las interacciones.
* El proyecto se crea mediante la aplicación y debe ser el líder inicial quien maneja esto.
* El líder debe manejar todo el proyecto, agregando o quitando actividades y los integrantes de este.
* Los integrantes deben tener actividades asignadas, las cuales deben cumplir individualmente o en grupo según la asignación.

Iteracion 3:

Con la iteración dos, logramos identificar algunos aspectos que nos faltan.

* Loader: Information holder. Es aquel que se encarga de extraer todos los datos previos de un proyecto.
* Aplicación: Director. Recordemos que este es quien maneja todas las relaciones y cambios de información entre los objetos del proyecto.
* Líder: Coordinador. Este puede mandar ordenes y hacer cambios en todo lo que tenga que ver con el proyecto que se maneja.
* Proyecto: Data holder. Esto es porque solo se limita a guardar y pasar información según lo pida la aplicación.
* Integrante: Data holder. Este objeto guarda la información de un integrante y la pasa o guarda según se requiera.
* Actividad: Data holder. Este objeto guarda todo lo que puede hacer una actividad.
* Tipo de Actividad: Data holder. Sera aquella parte que se definirá el rumbo que tomara la actividad que se esta desarrollando.
* Registro de Actividad: Data holder. Es una lista que almacena todos los cambios hechos por cada uno de los integrantes, de manera que sea fácil verificar quien hizo cada cambio.

Y con esto hecho, podemos actualizar y descomponer mas las responsabilidades e incluso distribuirla entre los objetos presentes.

* Loader:
  + Iniciar un proyecto nuevo.
* Aplicación:
  + Cargar la informacion previa del proyecto y de los integrantes.
  + Contar el tiempo transcurrido desde el inicio del proyecto.
  + Mostrar las opciones disponibles para cada usuario.
  + Agregar actividades para ser realizadas.
  + Crear y almacenar un registro de actividad.
  + Guardar información sobre lo que hizo cada uno de los integrantes.
  + Mostrar cual integrante ha trabajado mas en el proyecto.
  + Finalizar un proyecto.
  + Agregar integrantes.
  + Enviar registros al punto correspondiente según quien lo realizo.
* Lider:
  + Definir las actividades que deben realizarse.
  + Marcar las actividades que ya fueron realizadas.
  + Definir los plazos de tiempo para cada actividad.
  + Asignar el rol que ocupara un integrante (Líder o integrante normal).
* Integrante:
  + Marcar las actividades que ya fueron realizadas.
  + Mostrar cuantas horas se ha trabajado en una actividad.
* Actividad:
  + Definir el tipo de actividad.
* Registro de Actividad:
  + Almacenar datos (Registros) de lo hecho en el proyecto.
* Registro:

Requerimientos Funcionales:

- Cargar toda la información del proyecto: Se necesita que cada vez que se corra la aplicación, todos los datos se mantengan sin importar si se cerro o no..

- Crear un nuevo proyecto: La idea es que aquí se especifiquen aspectos básicos del proyecto en general, como por ejemplo, el nombre, el líder, integrantes, etc.

- Añadir integrantes o lideres: Se necesita que el tamaño de integrantes y lideres sea variable, pues estos proyectos pueden requerir mucha colaboración. Además, se necesita poder cambiar entre un integrante o un líder.

- Mantener un seguimiento del proyecto: Se requiere poder manejar de manera mas simple todo lo que gira en torno al proyecto: avances dia por dia, tiempo estimado y transcurrido, y todas las actividades que se necesiten realizar.

- Se requiere poder crear y asignar actividades a un integrante o líder dado: Ya que para resolver un proyecto se necesita ir paso por paso, es necesario que cada integrante tenga claro que papel debe desempeñar.

- Mantener un registro sobre todo lo hecho por cada integrante: Para facilitar el manejo de la aplicación y los cambios se necesita saber quien y cuando hizo cada actividad y cuanto tiempo ha trabajado diariamente,

Colaboraciones:

Dado que ya tenemos claro que papel desempeñara cada objeto, podemos hacer unas estimaciones sobre las relaciones que deben haber entre cada elemento.

* Aplicación: Posee relación con todos los objetos porque es aquel que se encarga de controlar el funcionamiento, comportamiento y actualización de cada uno de los elementos.
* Proyecto:
  + Tiene una relación con Aplicación: Aplicación será quien actualice toda la información en el proyecto y asignara las características de este.
  + Tiene una relación con líder: El proyecto necesita ser coordinado, y es el líder aquel que tiene control sobre todo lo que tenga que ver con el proyecto.
  + Tiene una relación con los integrantes: El proyecto es aquel que define cada cosa que debe ser realizado por los integrantes y les provee dicha información.
  + Tiene relación con Actividad: Para que el proyecto se encuentre finalizado, se requiere que se hayan cumplido una cierta cantidad de actividades y es por esto que el proyecto esta compuesto de una serie de actividades que el va creando.
  + Tiene relación con registro de actividad ya que el registro de actividad es una parte del proyecto, ahí es donde de almacena todos los cambios que surgen en el proyecto.
* Líder:
  + Tiene una relación con proyecto: El líder se encarga de gestionar todos los aspectos del proyecto.
  + Tiene una relación con los integrantes: Se encarga de asignar que actividades deben cumplirse y puede asignarla a un integrante especifico. Además, tiene la habilidad de cambiar a un integrante a líder.
  + Tiene relación con las Actividades: El es quien puede decidir como van a ser las nuevas actividades. -El las define, pero no las crea-
* Integrante: Aparte de las anteriores ya mencionadas, tiene una relación con las actividades; esto porque el integrante es aquel que las completa y puede dar una actividad por realizada.
* Actividad: Aparte de las relaciones ya especificadas, tiene una relación con Registro de actividad; esto debido a que Registro de actividad se encarga de mantener el historial de avances que se hicieron en cada actividad, esto con datos como la fecha de inicio, fecha de fin, tiempo transcurrido, persona que trabajo, etc.
* Registro de actividad: Todas sus relaciones ya fueron especificadas.

Reglas:

* La aplicación debe ser quien regule todas las interacciones.
* El proyecto se crea mediante la aplicación y debe ser el líder inicial quien maneja esto.
* El líder debe manejar todo el proyecto, agregando o quitando actividades y los integrantes de este.
* Los integrantes deben tener actividades asignadas, las cuales deben cumplir individualmente o en grupo según la asignación.
* Se necesita que haya un registro de todas las actividades dispuestas en el proyecto.

Iteracion 4:

Tras un análisis, nos pusimos a ver que mas cosas podrían faltar y nos dimos cuenta que no solo necesitamos algo que nos ayude a leer los datos guardados, sino que necesitábamos algo que escribiera nuevos datos y actualizara.  
  
Roles:  
fileWriter: Information holder– Entrara en contracto con todos los aspectos del proyecto y se encargara de almacenarlos. Se encontrara en el paquete de control.

Relaciones:  
Aparte de las que ya hemos visto con anterioridad, podemos sumar las nuevas.  
  
fileWriter: Entrara en contacto con el controlador de la aplicación, quien le pasara todos los datos necesarios para que se puedan almacenar en los .txt.

Responsabilidades nuevas:

Actividad: Contar tiempo en el cual fue ejecutada, esto gracias a los registros que hacen estos cálculos. Y también, la capacidad de ser trabajada por el usuario el tiempo que quiera

Integrante: Tendra la posibilidad de a través de una interfaz decidir que va a hacer. Este es capaz de trabajar en actividades, modificarlas y revisar en cualquier momento cuales le hacen falta revisar.  
  
Proyecto: Ahora, tiene la capacidad de transmitir datos al fileWriter, de tal manera que cada actualización que se haga se podrá cargar al iniciar y guardar al finalizar cada turno.

Registro de actividad: Ahora es el encargado de mantener todo lo que tenga que ver con las estadísticas y trabajo de cada integrante.

Registro: Es el encargado de tomar tiempo, fechas y asistencia de cada uno de los integrantes.

Colaboraciones:  
  
En su mayoría podemos ver que todas las clases tuvieron que ajustarse de una u otra manera al funcionamiento del lector y escritor de archivos.  
  
Es así como surgen las siguientes conexiones  
  
Registro de actividad y fileWriter: Uno recopila información mientras el programa se ejecuta para posteriormente pasar al fileWriter al final de la ejecución.  
  
Actividad:

* + Controlador: Se relacionan porque el controlador se encarga de almacenar en uno de sus atributos la actividad que se este desarrollando en ese momento
  + Aplicación: La actividad es parte vital de la aplicación, pues esta se maneja mucho en operar la actividad, obviamente su conexión no es directa pero una actividad es una parte vital de la aplicación
  + Usuario: El usuario es quien se encarga de decidir que se hace con esta de manera indirecta.

La aplicación tuvo muchos cambios importantes una vez inicio la implementación de la consola, pues este empezó a ser quien permitia que cada cosa se pudiera relamente conectar y ejecutar.

Metodos a implementar:  
  
Getter generales: Son necesarios para cada una de las clases que hay en la aplicación, desde las actividades hasta los usuarios.  
  
Funciones put: Se requerirán funciones Put, para que estas puedan definir sus atributos de manera totalmente directa (Esto esta enteramente pensada para la carga de datos.)  
  
getList(): Funciones necesarias tanto para obtener las actividades pendiente como los usuarios que pertenecen a un grupo. Consideramos que necesitamos bastantes de estas en cada clase.

actualizarTiempo(): Necesitamos que esta función sea capaz de internamente crear tiempos con base a fechas y subirlos al registro o actualice tiempos ya existentes en algunas actividades.

