

Entrega SIA AVANCE

Tema: Gestión de Agendas y Reuniones

Integrantes: Bastián Carrasco, Matias Fuentes y Jan Houter

1.1 Realizar un análisis de los datos a utilizar y principales funcionalidades a implementar que dan sentido a la realización del proyecto

Para el proyecto nuestro grupo seleccionó el tema de gestión de agenda y reuniones. En el cual se propone diseñar una aplicación para el tema escogido. Busca ser una mejora en el proceso de planificar y organizar el día a día, ya que las gestiones de agendas actuales son de un difícil manejo para los usuarios y no han sido modificadas a lo largo del tiempo. El objetivo de la aplicación es que el usuario pueda diseñar, organizar y planificar sus reuniones o actividades en una agenda de manera agradable y eficiente.

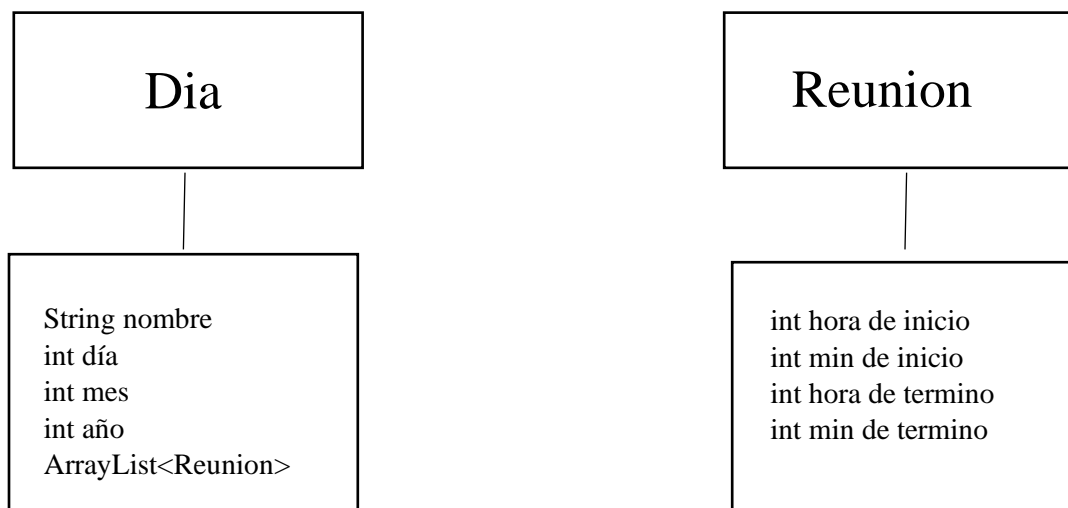
Para lograr lo anterior mencionado, se proponen las siguientes clases:

- Menú, clase principal de la aplicación donde se encuentra el menú.
- Día, clase que contendrá las fechas de las actividades.
- Reuniones, el usuario podrá ver las actividades programadas.

Funcionalidades que dan sentido al proyecto:

- mostrar, llama al método data_toda.
- data_toda, muestra los datos de la actividad/es de la persona.
- seguridad, hace validaciones correspondientes.

1.2 Diseño conceptual de clases del Dominio y su código en Java



1.3 Todos los atributos de todas las clases deben ser privados y poseer sus respectivos métodos de lectura y escritura (`getter()` / `setter()`)

Se consideraron las clases Día y Reuniones como se pueden ver en los archivos java correspondientes se tienen los atributos en privado junto con su correspondiente `getter` y `setter`.

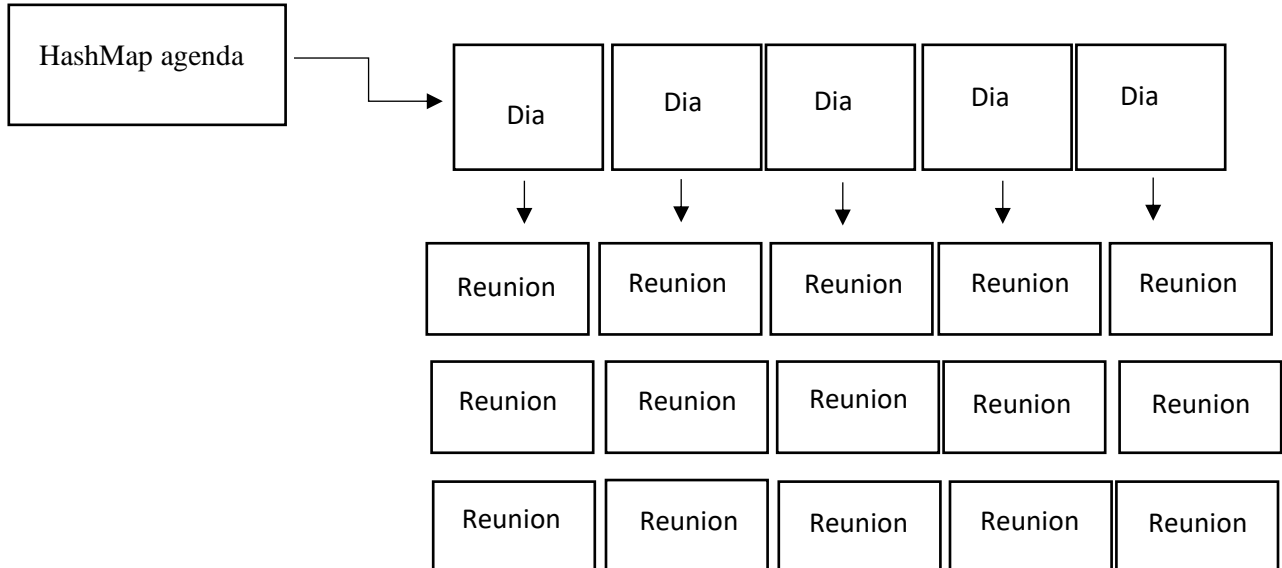
En el archivo `Dia.java` podemos ver entre las líneas 16 y 20 como los atributos están en privado y entre las líneas 31 y 79 los métodos `getter` y `setter` correspondiente.

En el archivo `Reuniones.java` podemos ver entre las líneas 14 y 18 como los atributos están en privado y entre las líneas 29 y 67 los métodos `getter` y `setter` correspondiente.

1.4 Se deben incluir datos iniciales dentro del código

Los datos iniciales se encuentran dentro de la clase `AgendayReuniones`, en el `main`. Existe un método llamado `llener_con_datos` y se puede visualizar entre las líneas 198 y 244.

1.5 Diseño conceptual y codificación de 2 colecciones de objetos, con la 2ª colección anidada como muestra la figura. Las colecciones pueden ser implementadas mediante arreglos o clases del Java Collections Framework (JCF)

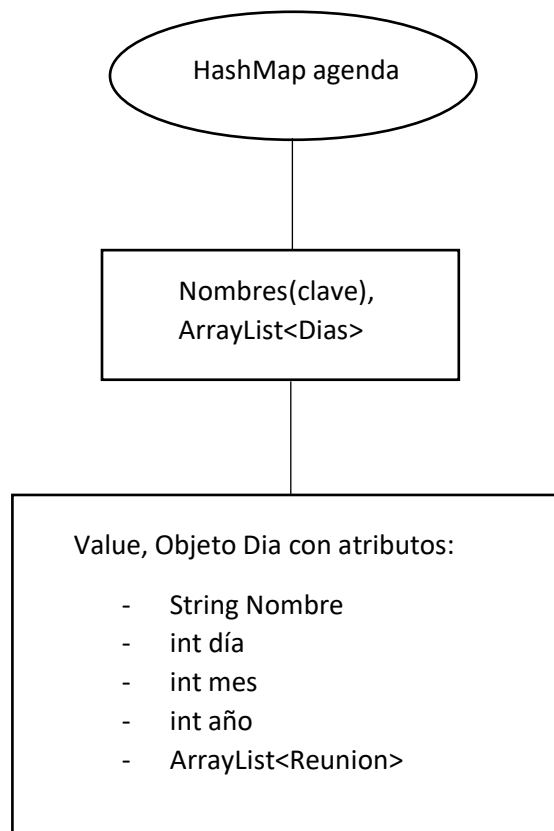


1.6 Diseño conceptual y codificación de 2 clases que utilicen sobrecarga de métodos (no de constructores)

En la clase Días entre las líneas 104 y 114 se encuentra el método seguridad, el cual valida que el día y mes de la actividad ingresada se encuentren dentro de los parámetros determinados. Y a su vez se encuentra seguridad, como sobrecarga de método el cual valida que el año de la actividad se encuentre valido.

Y en la clase Reunión entre las líneas 85 y 96 se encuentra el método seguridad, el cual valida que la hora, minuto inicial como final se encuentren dentro de los parámetros determinados. Y a su vez se encuentra seguridad, como sobrecarga de método el cual valida que la hora final no sea menor que la hora inicial.

1.7 Diseño conceptual y codificación de al menos 1 clase mapa del Java Collections Framework



Como se aprecia en la figura, nuestro `HashMap agenda`, trabaja con un objeto value `Dia` con atributos propios y es identificado a partir de la clave nombre.

1.8 Se debe hacer un menú para el Sistema donde ofrezca las funcionalidades de:
1) Inserción Manual / agregar elemento y 2) Mostrar por pantalla listado de elementos. Esto para la 2ª colección de objetos (colección anidada) del SIA1.5

Al momento de ejecutar se puede apreciar el menú con las funcionalidades solicitadas. En la clase AgendayReuniones, en el método crearMenu, se imprime por pantalla el menú. Y entre las líneas 186 y 195 se puede apreciar como el menú fue programado para su funcionamiento.

1.9 Todas las funcionalidades deben ser implementadas mediante consola (Sin ventanas)

Se implementaron las funciones solicitadas por consola.

1.10 Utilización de GitHub (Realización de al menos 3 Commit)

Como se puede ver en el historial de repositorio se hicieron los commit correspondientes.