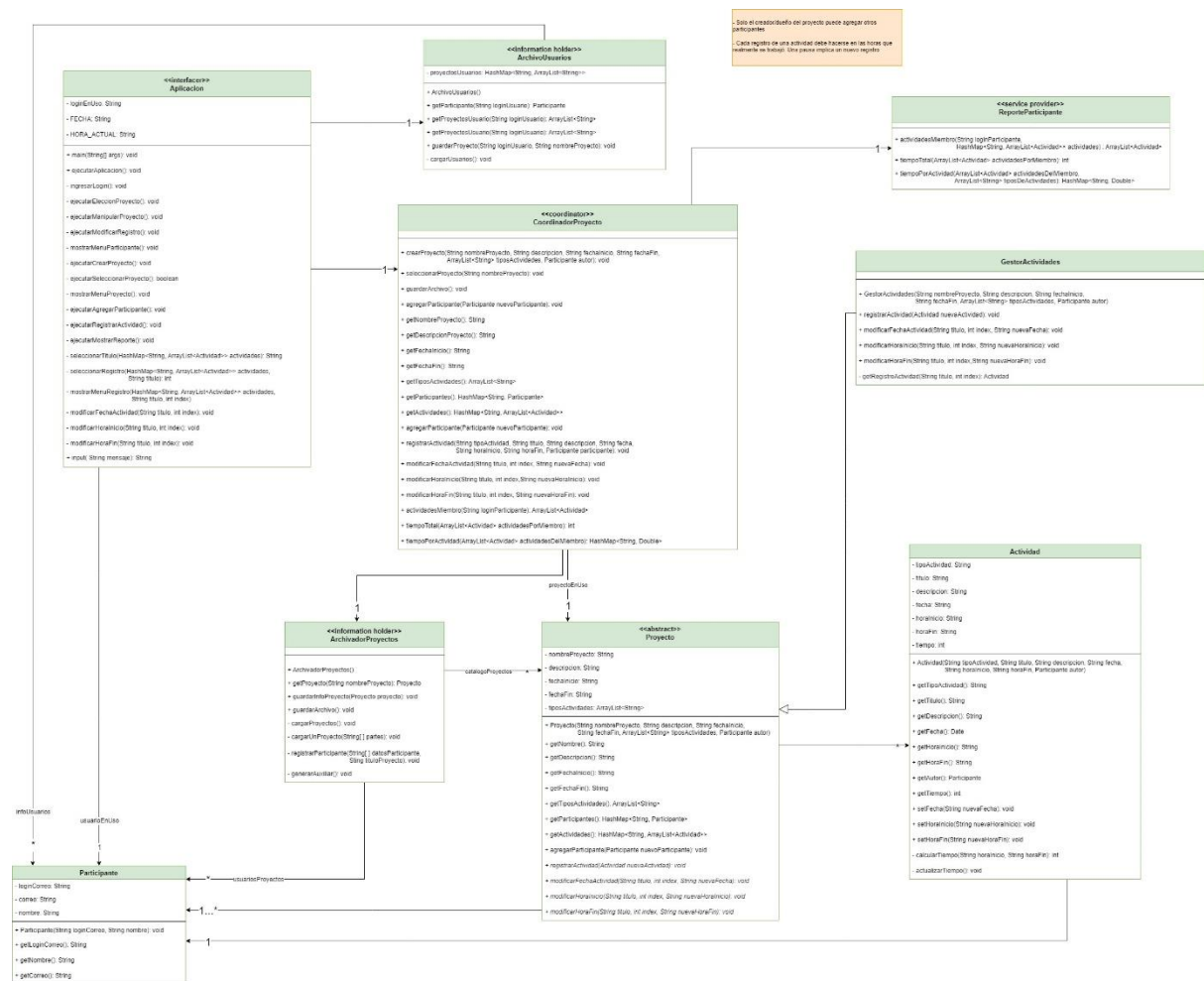


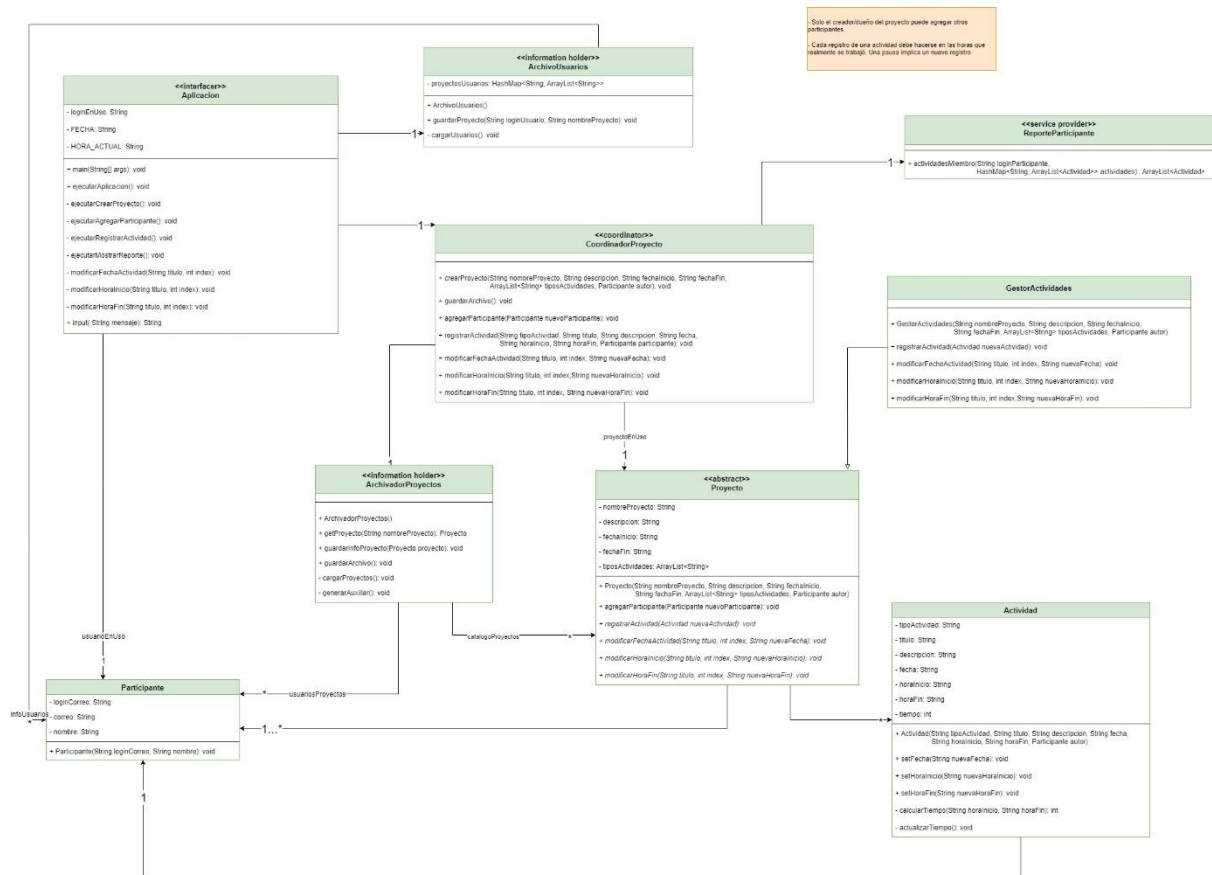
Juan Sebastián Sánchez 202013577

**Diagrama de clases:**



**(Dentro de la misma carpeta de Github Se encuentra este mismo diagrama en PDF para más claridad)**

**Diagrama de clases alto nivel:**



(Dentro de la misma carpeta de Github Se encuentra este mismo diagrama en PDF para más claridad)

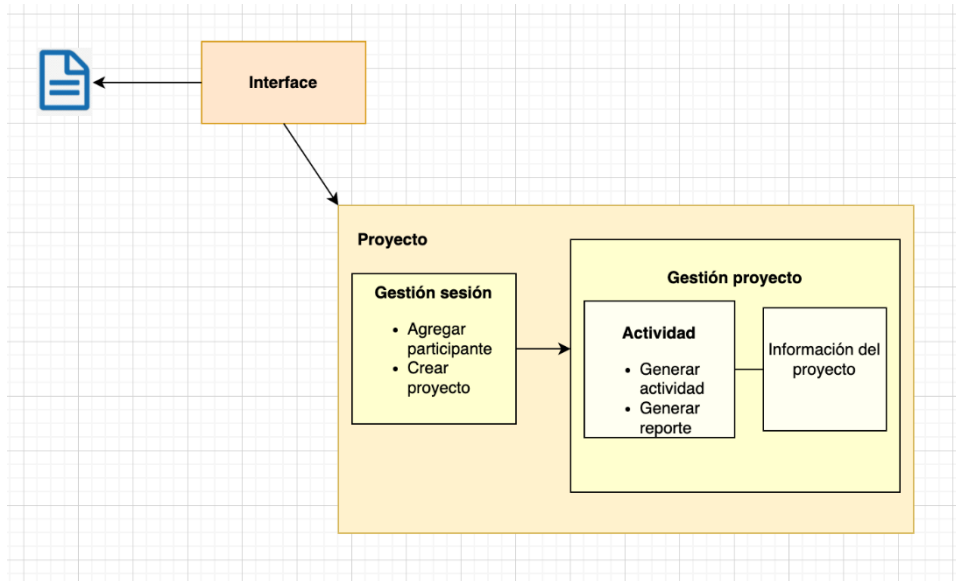
## Explicación Diseño:

El diseño del programa empezó de dos formas simultaneas. La primera de estas se basaba en imaginarse como seria la interacción entre el usuario y el programa. Esto dio como resultado un enfoque relativamente grande en un componente de interfaz para poder mediar la interacción de la aplicación con el usuario. Adicionalmente, ayudo a estructurar la jerarquía de menús, lo que al mismo tiempo permitiría una aproximación a la jerarquía o el orden en términos de un diagrama de clase.

Esta interacción entre el usuario y la aplicación se basaría en una interfaz por consola. Esta interfaz muestra las opciones disponibles de las acciones, navegación dentro de la aplicación y posibilidad de cerrar el programa. Dicha interfaz primero empieza preguntando un login para permitir el acceso a un usuario. A raíz de esto, es necesario que la interfaz sepa que usuarios están registrados para revisar si un login ingresado hace parte esta dentro de los archivos de la aplicación, y si no, que pueda registrar un usuario. Posteriormente, al iniciar sesión con como un usuario, debe permitir acceder a los proyectos a los cuales hace parte o debe de poder crear un proyecto, lo que implica que el mismo archivo o information holder que tiene los usuarios registrados, debe tener acceso o por lo menos conocer los nombres de los proyectos a los cuales hace parte un usuario. Con estos datos, se procede entonces a pasar a un menú o una sección del proyecto diferente.

El siguiente menú consiste en el menú de acciones de proyecto, en donde se pueden registrar nuevos participantes en el proyecto actual, generar registros de actividad, modificar registros de actividad y generar reportes de actividades para los participantes del proyecto. Esta organización, para su implementación, requería de la diferenciación entre un elemento que coordine los entes y operaciones que se requieren en esta etapa. Operaciones tales como la generación de los reportes, la generación de actividades y su modificación y los usuarios escritos en el proyecto con su información.

Al mismo tiempo, se empezaron a realizar bosquejos de diagramas de alto nivel que describan los componentes que se podrían utilizar en la solución



En este primer bosquejo, se empieza con un interface con acceso a información. Esta información era lo que se mencionó acerca de los participantes registrados como usuarios y los proyectos, o nombres de proyectos a los cuales pertenecen. Luego, se entra a un paquete o grupo que consiste en más divisiones. Una gestión de sesión, que se encarga de agregar participantes a los datos de la aplicación y la creación de proyecto. Este paquete entra en contacto con una gestión de proyecto, la cual es la que se encarga de generar actividades de los proyectos y generar sus reportes. Adicionalmente, también contiene la información del proyecto que se refiere a datos tales como descripción del proyecto y los tipos de actividades que soporta.

Posteriormente, revisiones de este primer diagrama revelaron ciertos errores y mejoras que se podrían implementar. Por ejemplo, la gestión de sesión podía estar aparte del elemento que tiene que ver el proyecto, ya que esta no tiene que ver mucho internamente con los elementos conceptuales del proyecto. Más bien, que la gestión de sesión se pueda manejar en conjunto con la interfaz, y que sea la interfaz la que medie las interacciones entre sesión y los proyectos. Otra modificación necesaria es pasar la responsabilidad de creación de proyecto y agregar participantes a algo que no sea la gestión de sesión. Estas revisiones iniciales poco a poco empezaron a dar fruto a los diagramas de clases mostrados al inicio del documento.

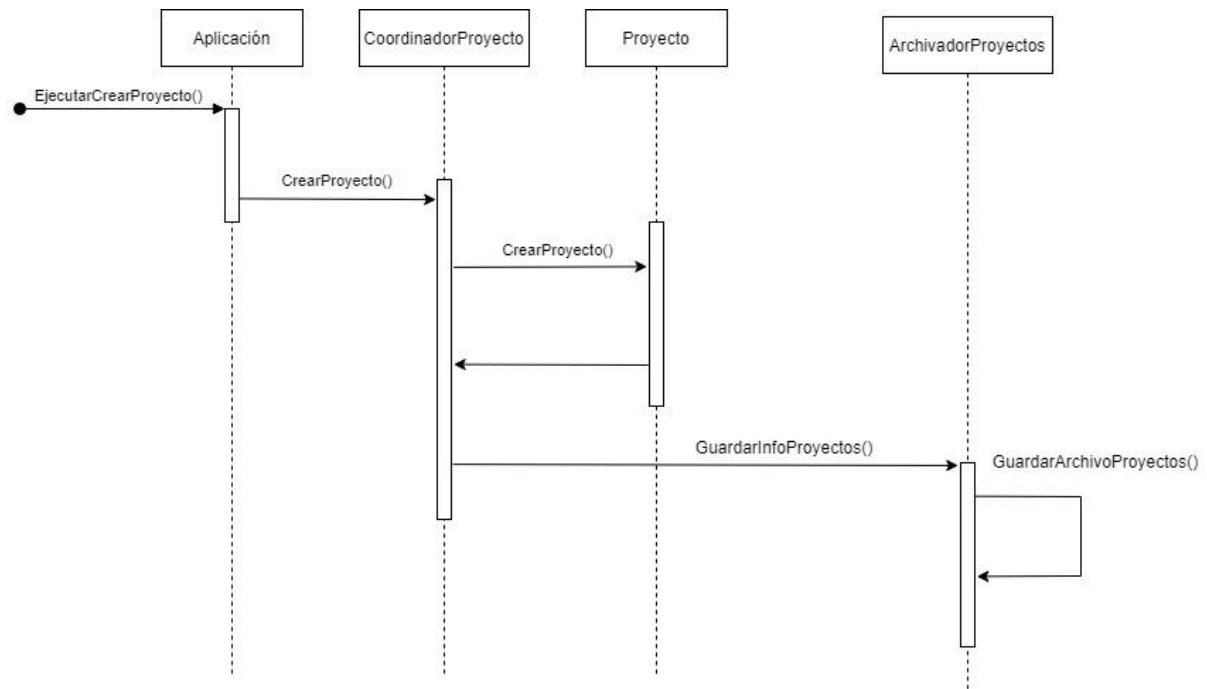
Posteriormente, revisiones realizadas al mismo tiempo que se realizaba la implementación, entre estas consideraciones nació la necesidad de cambiar la clase proyecto, que anteriormente era una sola instancia, en una clase abstracta que sea implementada por un gestor de actividades. Esto para que, un gestor de actividad debe saber la información del proyecto, o contenerla para poder hacer la generación de actividades y sus posibles modificaciones. Y un elemento adicional que genere los reportes por participante.

Una clase que cambió considerablemente durante la implementación del problema fue `archivadorProyectos`. Responsable de almacenar el catálogo de proyectos, además de implementar funciones para llamar un proyecto y guardar información de los proyectos creados y modificados. Otra responsabilidad de la que se encarga esta clase es la carga del archivo fuente. Este archivo fuente contiene la información de los proyectos, sus participantes y actividades. Al leer este documento, crea los proyectos, adjunta sus participantes a los proyectos y sus actividades, a la vez que genera un archivo auxiliar que posteriormente, la clase `archivo usuarios` utilizara para reconocer que usuarios están registrados.

### Diagramas de secuencia para métodos críticos:

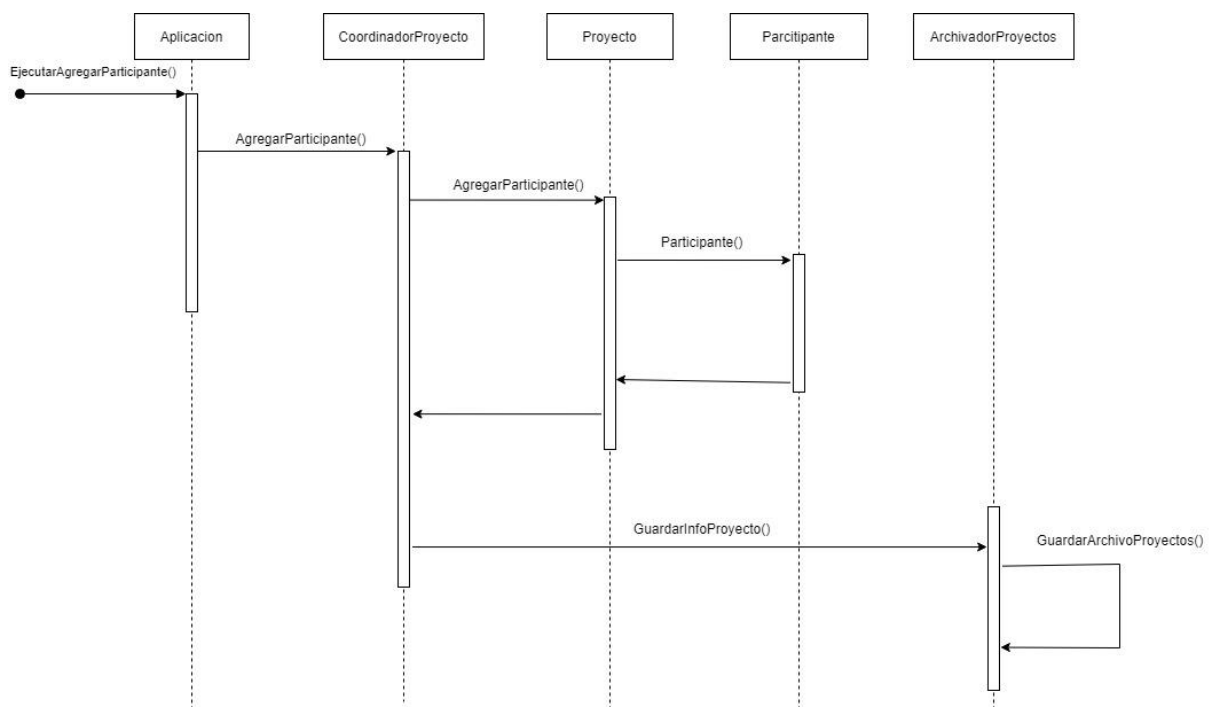
Los métodos críticos del proyecto se consideraron como los que consisten en realizar la creación de ciertos requerimientos. Entre estos, se consideraron los métodos de crear proyecto, añadir participantes y el registro de actividades. Estos diagramas son los siguientes:

Crear un Proyecto:



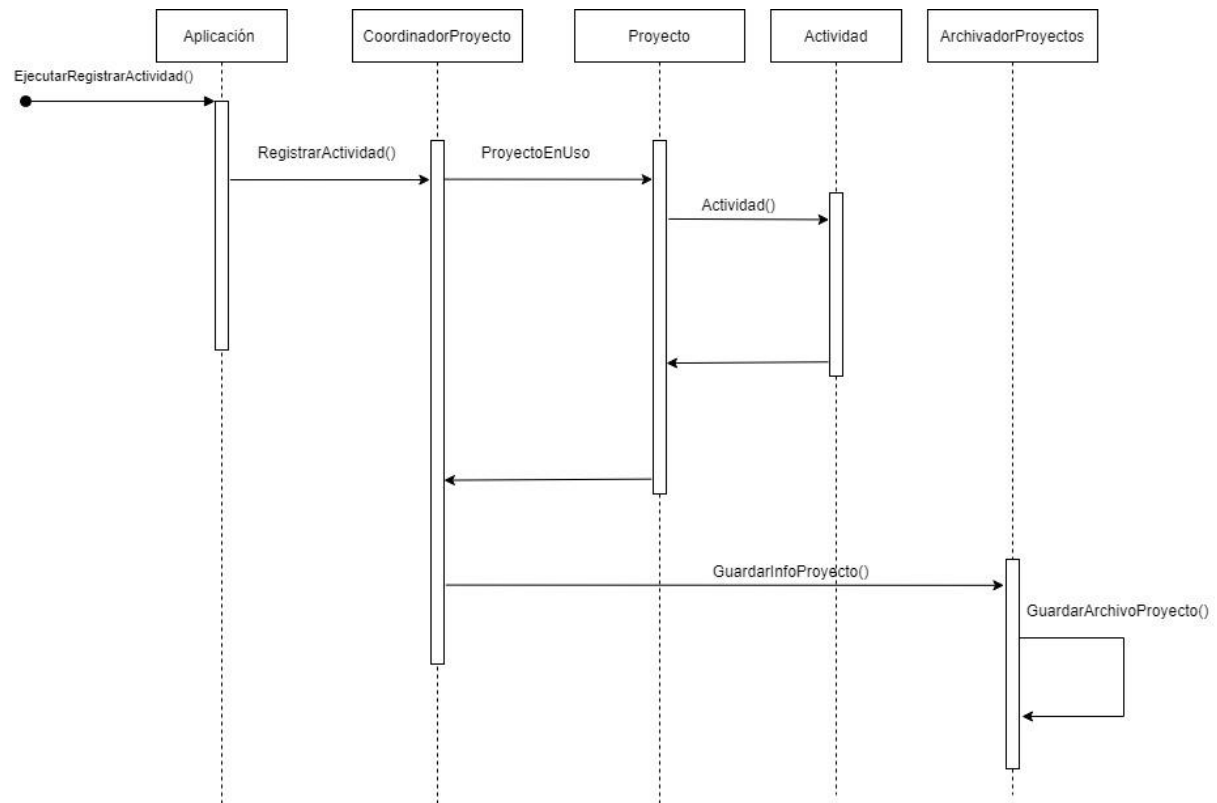
Durante la ejecución de este método, se asume que anteriormente ya se inicio un usuario el cual es el que esta creando el proyecto. Este usuario no se inicializa como un objeto o algo parecido dado a que se tiene almacenado como un atributo que representa que es el usuario actual.

### Agregar participante



En este diagrama de secuencia, también sucede algo similar que en el anterior. Ya se está con un participante. En este caso, lo que sucede es que se acciona el método para agregar un participante dentro del proyecto, esto posteriormente llama a la clase participante para poder crearlo y anexasarlo al proyecto.

### Registrar Actividad:



La estructura de este diagrama de secuencia sigue una lógica similar al anterior. En donde ya se tiene un participante que está realizando el registro.

Algo en común que comparten todos los diagramas de secuencia es la particularidad de que al final de registrar una creación de nueva información del proyecto es que se guarda la info del proyecto y posteriormente se guarda o actualiza la información reportada en un archivo de proyecto. Esto se hace de forma que, después de agregar cosas en proyectos, se puedan registrar dichos cambios.