Imagen que contiene Diagrama

Descripción generada automáticamente

Universidad de los Andes

Ingeniería de Sistemas y Computación

Diseño y Programación Orientada a Objetos

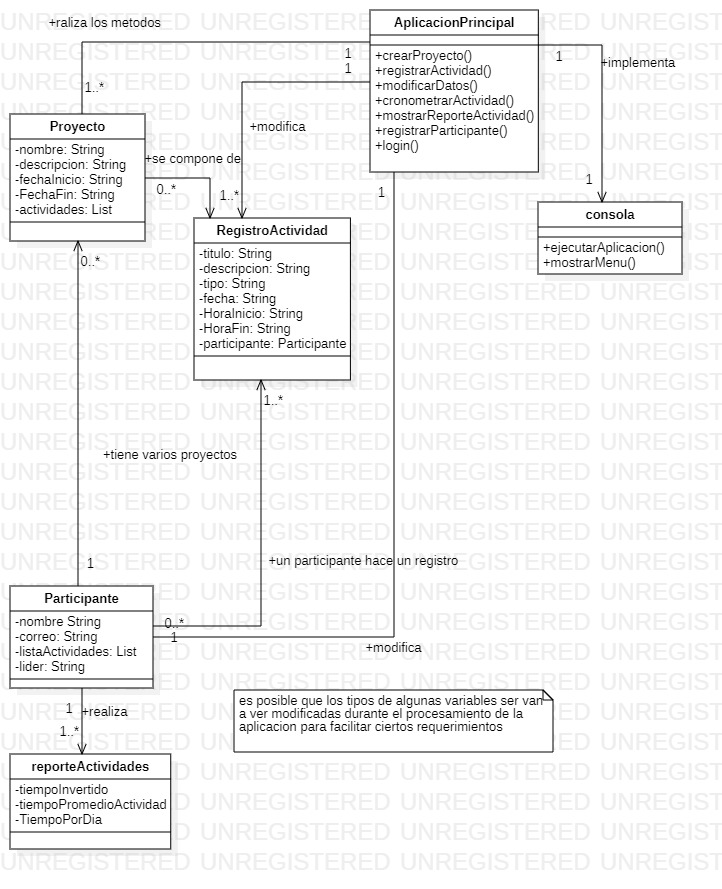
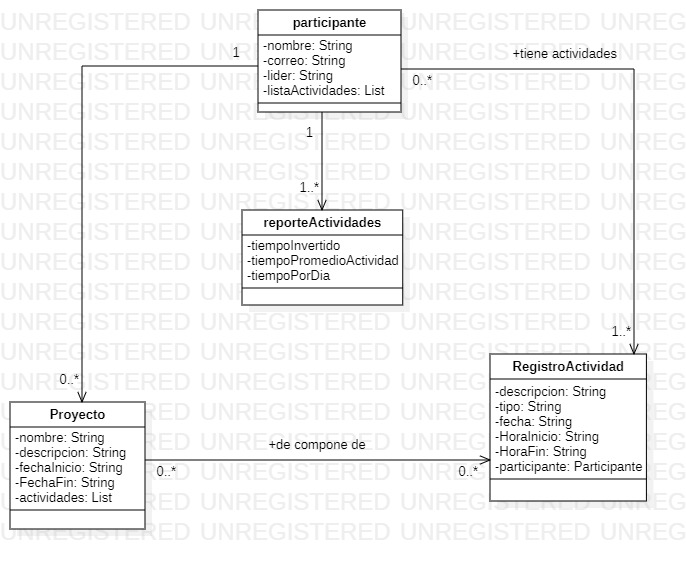
**Proyecto1**

Entrega 2- Documento de diseño

Juan Diego Cruz Vega

[j.cruzv@uniandes.edu.co](mailto:j.cruzv@uniandes.edu.co) 202015077  
  
Mateo López Tobón  
[m.lopez24@uniandes.edu.co](mailto:m.lopez24@uniandes.edu.co) 202021416  
  
Mateo Cote Canal  
[m.cotec@uniandes.edu.co](mailto:m.cotec@uniandes.edu.co) 202022609

Bogotá, Colombia

****

**Principales:**

1. Proyecto:
2. Participante:
3. RegistroActividad:

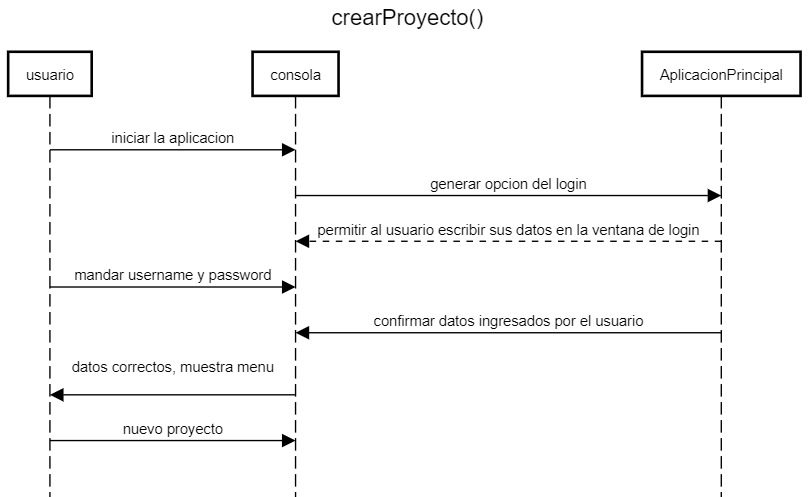
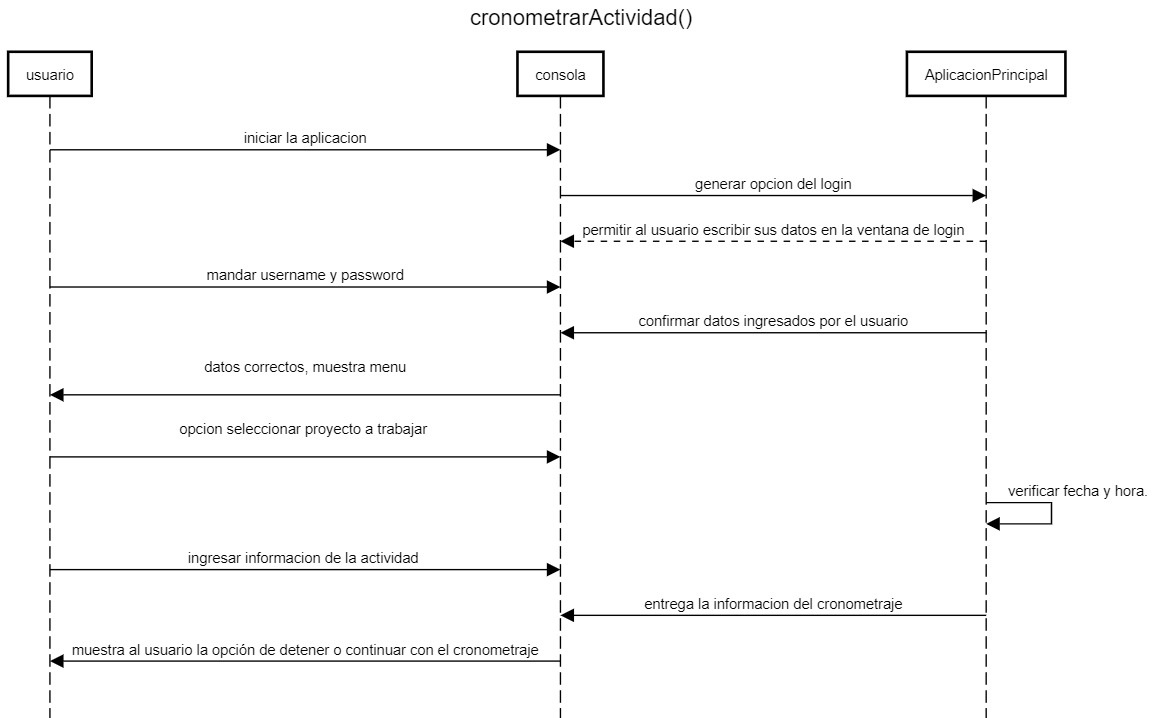
**Relaciones entre los objetos: (según los diagramas podemos describir las relaciones entre cada clase)**

* **RegistroActividad->proyecto** : Proyecto este compuesto por actividades y del mismo modo puede ser modificado por estas, en proyecto puede haber una o muchas actividades.
* **aplicacionPrincipal–proyecto**: Por medio de la app principal se agregan proyectos y le ejecuta las funciones a este, por ejemplo, se le registran actividades a un proyecto específico.
* **AplicaciónPrincipal–>consola:** La aplicación principal implementa la consola, a través de esta se ejecuta la aplicación y se muestra el menú.
* **AplicaciónPrincipal–>registroActividad:** Por medio de la app principal se modifican y agregan los datos del registro de una actividad.
* **Participante–>proyecto:** Cada participante está en uno o más proyectos, haciéndolo proyecto un atributo de la clase participantes.
* **Participante–reporteActividades:** Los participantes realizar los reportes del día, en el cual se registra el tiempo diario invertido en las actividades.
* **Participante–AplicacionPrincipal:** A través de la app principal se registran y modifican los datos de la clase participante.
* **Participante–>RegistroActividad:** Al igual que en reporteActividades, el participante realiza un registro de actividad, y provee los datos necesaria para esta.

**Métodos empleados:**

1. **crearProyecto():** a través de la app principal se crea uno o varios objetos de la clase Proyecto, y se le dan sus valores respectivos.
2. **registrarActividad():** a través de la app principal se crea uno o varios objetos de la clase registrarActividad, y se le dan sus valores respectivos.
3. **modificarDatos():** a través de la app principal se modifican los datos existentes de una actividad o participante.
4. **cronometrarActividad():** a través de la app principal se ejecuta la función que cronometra el tiempo gastado por un participante en una actividad especifica.
5. **mostrarResporteActividad():** dados los datos recolectados en los métodos anterior, la app principal crea un reporte con la información requerida de cierta actividad.
6. **registraParticipante():** a través de la app principal se crea uno o varios objetos de la clase Participante, y se le dan sus valores respectivos.
7. **login():** El usuario inicia sesión para poder realizar cualquiera de los métodos anteriores.

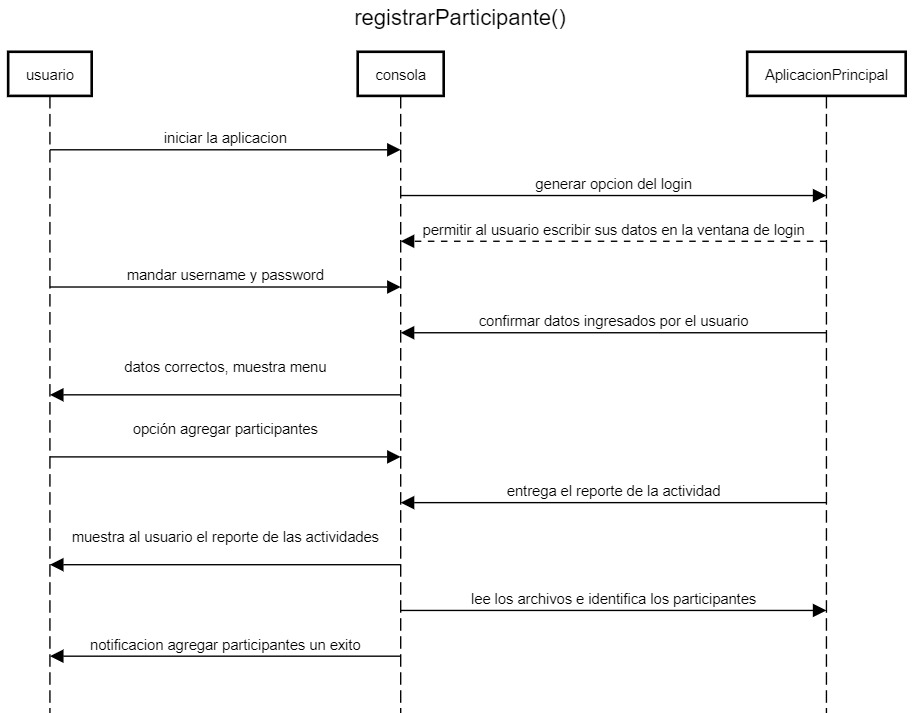
**Diagramas de secuencia**

* crearProyecto():  
  
* cronometrarActividad():   
  
* login()  
  Diagrama

  Descripción generada automáticamente
* modificarDatos()  
  Diagrama

  Descripción generada automáticamente
* mostrarReporteActividad()Diagrama

  Descripción generada automáticamente
* registrarActividad()Diagrama

  Descripción generada automáticamente
* registrarParticipante()

**Reflexiones**

Con esto se concluye la planeación (análisis y diseño) del proyecto, con esto se espera tener una buena idea formulada para el modelo del proyecto, y dependiendo de las complicaciones en la implementación, podremos esperar un leve cambio en el desarrollo, pero en general que el diseño se mantenga coherente y muy similar con esta versión presentada.

Se espera que con este diseño se pueda desarrollar de la forma más eficiente lo necesario para abarcar lo requerido en este proyecto y poder hacerlo en el tiempo deseado.