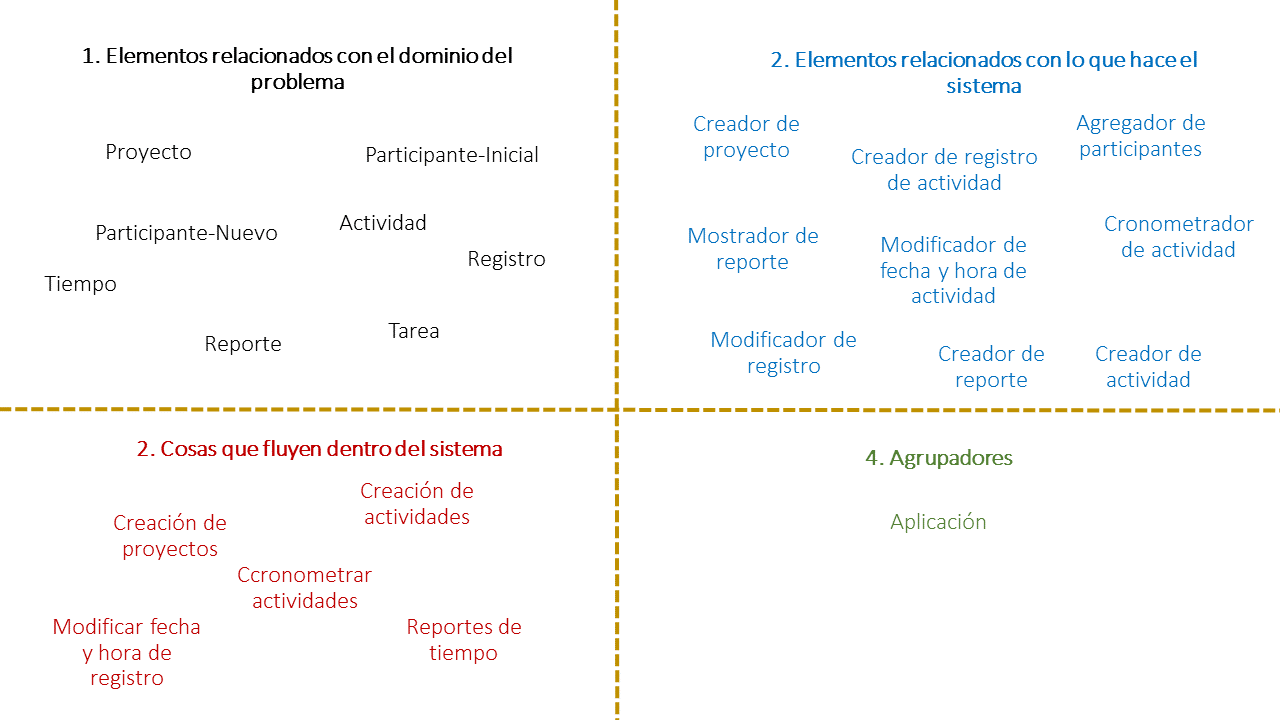
# Proceso de diseño

Hay que basarnos en el [Documento de análisis](https://docs.google.com/document/d/1wHE7zh3D85-x80IzTJVJndD6fxquvQ65h7Oeg8ppMnk/edit) que hicimos en la primera entrega.

## Primera iteración

### Roles



### Roles

Al analizar los elementos relacionados con lo que hace el sistema podemos darnos cuenta de algunos elementos que se pueden agrupar y otros que son necesarios para relacionar otros roles, de tal modo qué:

1. **Proyecto:** este rol será el encargado de almacenar toda la información de un proyecto creado por un *participante inicial.* El componente principal que debe contener el proyecto son las *actividades*.
2. **Participante:** este rol es clave, pues agrupa elementos como: “Agregador de participantes”, “Creador de proyecto”, “Creador de actividad” y “Creador de registro de actividad”. Sin embargo, es importante aclarar que de este rol participante se desencadenaran los siguientes roles heredados:

**2.1. Participante Inicial:** encargado de realizar en sí la creación del proyecto, también agrega participantes y puede registrar una actividad.

**2.2 Participante Nuevo:** también puede registrar una actividad.

1. **Actividad:** este rol almacena la información y tipo de una actividad realizada por un participante. Existirán varios tipos de actividades; “Documentación”, “Implementación”, “Pruebas”, “Investigación”, “Diseño”, etc.. que heredarán los atributos y métodos de Actividad.
2. **Registro:** este rol se encarga de almacenar los registros de una *actividad,* es útil para cumplir el elemento de “Modificador de Registro” en el cual se modifican tareas que se hicieron horas o días antes, donde no se tenía acceso al sistema.
3. **Reporte:** este rol se encarga de almacenar información de las *actividades* de **cada** *participante,* específicamente del tiempo de cada actividad (invertido, promedio, por cada día, etc). Es el “Creador de reportes”.
4. **Aplicación:** este rol se encarga de agrupar y relacionar algunos roles anteriores, como *Reporte*, *Registro, Participante* y *Actividad*, de manera que posee tareas como “Mostrador de reporte” y “Cronometrador de actividad”.

#### Actividades de toma de decisiones, control y coordinación.

|  |  |  |
| --- | --- | --- |
| **Toma de decisiones** | **Control** | **Coordinación** |
| Modificar fecha y hora de actividad. | Agregar participantes. | Crear reporte. |
| Modificar registro. | Crear registro. | Crear proyecto. |
| Cronometrar actividad. | Crear actividad. | Mostrar reporte. |

#### Roles basados en estereotipos:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Roles | Information holder | Structurer | Service provider | Coordinator | Controller | Interfacer |
| 1 |  | X |  |  |  |  |
| 2 |  |  |  |  | X |  |
| 3 | X |  |  |  |  |  |
| 4 |  | X |  |  |  |  |
| 5 |  |  | X |  |  |  |
| 6 |  | X |  |  |  |  |

### Responsabilidades

Para establecer responsabilidades, decidimos usar la estrategia que vimos en clase.

|  |  |  |
| --- | --- | --- |
| Saber algo | Tomar decisiones | Realizar acciones |
| Actividad | Aplicación | Participante |
| Reporte |  | Aplicación |
| Proyecto |  | Registro |

#### 

#### Responsabilidades del sistema

1. Crear proyecto
2. Agregar participantes
3. Crear registro de actividad
4. Modificar registro de actividad
5. Cronometrar actividad
6. Mostrar reporte de actividades

#### Responsabilidades del sistema implícitas

1. Realizar una colaboración de participantes en una actividad

#### Si pasa… entonces

##### Crear proyecto

* Si creo un proyecto debe haber al menos un participante con su nombre y correo (participante inicial).
* Si creo un proyecto debe tener un nombre, descripción, una fecha de inicio y una fecha estimada de finalización si ya se conoce.

##### Agregar participantes

* Si agrego un participante cada uno tener nombre y correo (participante nuevo).
* Si agrego un participante ya debe haber sido creado un proyecto con sus atributos.

##### Registrar actividad realizada

* Si registro una actividad realizada cada actividad debe tener un título, una descripción, un tipo, una fecha en la que se realizó, una hora de inicio, una hora de finalización y un participante que la realizó.

##### Modificar registro

* Si modifico el registro debe haber un registro de actividad a la cual pueda modificar sus atributos.

##### Cronometrar actividad

* Si voy a cronometrar una actividad debe existir una actividad.
* Si voy a cronometrar una actividad entonces puedo pausar y reanudar el cronómetro.

##### Mostrar reporte de actividades

* Si voy a ver el reporte de actividades, al menos debe existir una actividad.

#### Responsabilidades basadas en estereotipos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Responsabilidades | Information holder | Structurer | Service provider | Coordinator | Controller | Interfacer |
| 1 |  | X |  |  |  |  |
| 2 |  |  |  | X |  |  |
| 3 | X |  |  |  |  |  |
| 4 |  |  |  |  | X |  |
| 5 |  |  | X |  |  |  |
| 6 | X |  |  |  |  |  |

#### 

#### Otras responsabilidades técnicas necesarias

* Por el momento no encontramos ninguna otra.

### 

### Colaboraciones

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Responsabilidades | Proyecto | Participante | Actividad | Registro | Reporte | Aplicación |
| 1 | X |  |  |  |  |  |
| 2 | X | X |  |  |  |  |
| 3 |  | X | X |  |  |  |
| 4 |  | X |  | X |  |  |
| 5 |  |  |  |  |  | X |
| 6 |  |  |  |  | X | X |

* Proyecto y participante se agrupan con agregar participantes dado a que para agregar participantes debe existir ya un proyecto previamente y un participante para agregar.

* Actividad y participante se agrupan con registrar actividad ya que existe una relación directa porque no puede registrar una actividad si no existe una actividad y además es una función del participante.
* Registro y participante se agrupan con modificar registro ya que existe una relación directa porque no se puede modificar un registro si no existe un registro y porque es una función del participante.
* Reporte y aplicación se agrupan con mostrar reporte de actividades porque existe una asociación directa dado que se necesita de un reporte para mostrar el reporte de actividades y además es una función de la aplicación.

## Segunda Iteración

### Cambios añadidos:

#### Roles

1. No hay cambios

#### Responsabilidades

1. Realizar una colaboración de participantes en una actividad.

* Notamos que debe ser posible registrar una actividad realizada por alguien más dentro del proyecto. Por esto, creemos necesario crear esta colaboración entre participantes.

1. Crear actividad

* Es diferente crear una actividad que registrarla, anteriormente solo teníamos el registro de dicha actividad. Sin embargo, no se puede realizar un registro sin tener una actividad creada previamente.

1. Cronometrar actividad

* Es algo que el sistema hace automáticamente. Por lo tanto, es necesario cambiar el cómo planeamos esta responsabilidad, pues el participante del proyecto debe tener la posibilidad de pausar y continuar una actividad en dicho cronometraje.

1. Pausar cronómetro de actividad

* Como se dijo anteriormente, el participante que está interactuando con una actividad es el encargado de pausar el cronómetro.

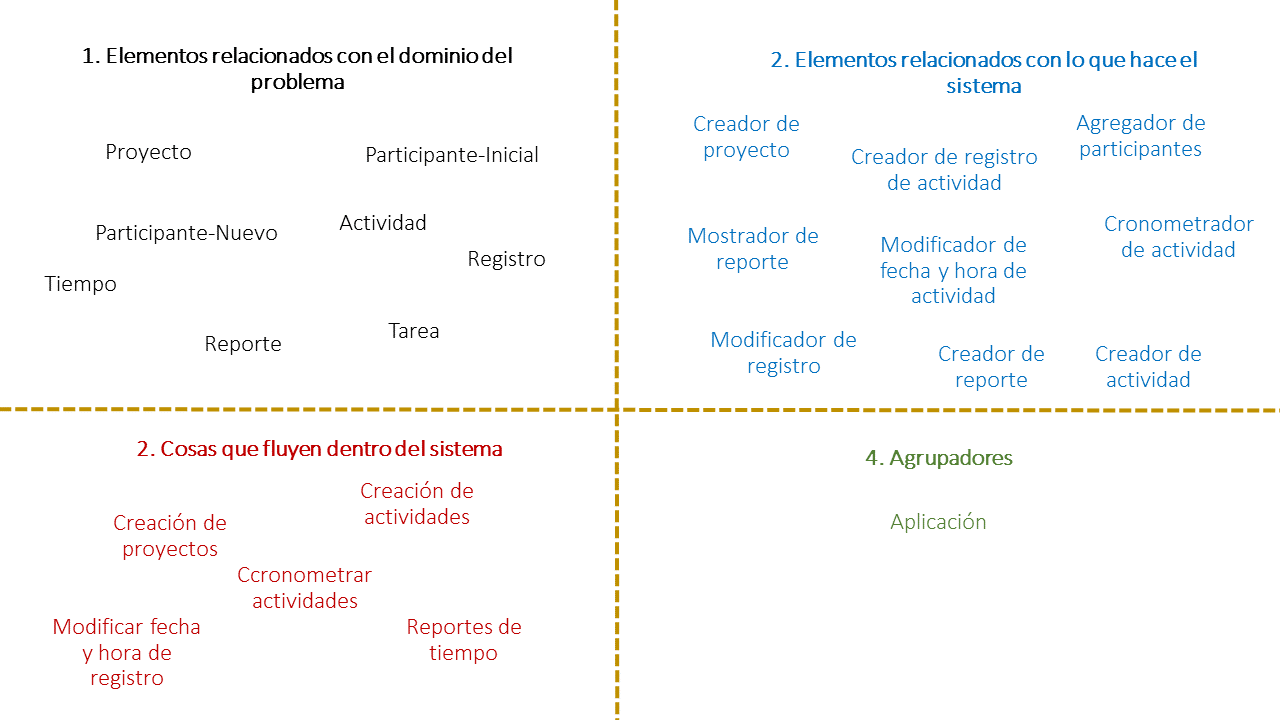
1. Reanudar cronómetro de actividad

* Como se dijo anteriormente, el participante que está interactuando con una actividad es el encargado de pausar el cronómetro.

1. Crear reporte

* Nos dimos cuenta de que para poder mostrar un reporte es necesario que antes de eso, se cree un reporte.

### Roles

Al analizar los elementos relacionados con lo que hace el sistema podemos darnos cuenta de algunos elementos que se pueden agrupar y otros que son necesarios para relacionar otros roles, de tal modo qué:

1. **Proyecto:** este rol será el encargado de almacenar toda la información de un proyecto creado por un *participante inicial.* El componente principal que debe contener el proyecto son las *actividades*.
2. **Participante:** este rol es clave, pues agrupa elementos como: “Agregador de participantes”, “Creador de proyecto”, “Creador de actividad” y “Creador de registro de actividad”. Sin embargo, es importante aclarar que de este rol participante se desencadenaran los siguientes roles heredados:

**2.1. Participante Inicial:** encargado de realizar en sí la creación del proyecto, también agrega participantes y puede registrar una actividad.

**2.2 Participante Nuevo:** también puede registrar una actividad.

1. **Actividad:** este rol almacena la información y tipo de una actividad realizada por un participante. Existirán varios tipos de actividades; “Documentación”, “Implementación”, “Pruebas”, “Investigación”, “Diseño”, etc.. que heredarán los atributos y métodos de Actividad.
2. **Registro:** este rol se encarga de almacenar los registros de una *actividad,* es útil para cumplir el elemento de “Modificador de Registro” en el cual se modifican tareas que se hicieron horas o días antes, donde no se tenía acceso al sistema.
3. **Reporte:** este rol se encarga de almacenar información de las *actividades* de **cada** *participante,* específicamente del tiempo de cada actividad (invertido, promedio, por cada día, etc). Es el “Creador de reportes”.
4. **Aplicación:** este rol se encarga de agrupar y relacionar algunos roles anteriores, como *Reporte*, *Registro, Participante* y *Actividad*, de manera que posee tareas como “Mostrador de reporte” y “Cronometrador de actividad”.

#### Actividades de toma de decisiones, control y coordinación.

|  |  |  |
| --- | --- | --- |
| **Toma de decisiones** | **Control** | **Coordinación** |
| Modificar fecha y hora de actividad. | Agregar participantes. | Crear reporte. |
| Modificar registro. | Crear registro. | Crear proyecto. |
| Cronometrar actividad. | Crear actividad. | Mostrar reporte. |

#### Roles basados en estereotipos:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Roles | Information holder | Structurer | Service provider | Coordinator | Controller | Interfacer |
| 1 |  | X |  |  |  |  |
| 2 |  |  |  |  | X |  |
| 3 | X |  |  |  |  |  |
| 4 |  | X |  |  |  |  |
| 5 |  |  | X |  |  |  |
| 6 |  | X |  |  |  |  |

### Responsabilidades

Para establecer responsabilidades, decidimos usar la estrategia que vimos en clase.

|  |  |  |
| --- | --- | --- |
| Saber algo | Tomar decisiones | Realizar acciones |
| Actividad | Aplicación | Participante |
| Reporte |  | Aplicación |
| Proyecto |  | Registro |

Nota: Las resaltadas en rosado son las nuevas responsabilidades.

Responsabilidades totales:

1. Crear proyecto
2. Agregar participantes
3. Crear actividad
4. Crear registro de actividad
5. Modificar registro de actividad
6. Crear reporte
7. Mostrar reporte de actividades
8. Realizar una colaboración de participantes en una actividad
9. Cronometrar actividad .
10. Pausar cronómetro de actividad
11. Reanudar cronómetro de actividad

#### Si pasa… entonces

##### Crear proyecto

* Si creo un proyecto debe haber al menos un participante con su nombre y correo (participante inicial).
* Si creo un proyecto debe tener un nombre, descripción, una fecha de inicio y una fecha estimada de finalización si ya se conoce.

##### Agregar participantes

* Si agrego un participante cada uno tener nombre y correo (participante nuevo).
* Si agrego un participante ya debe haber sido creado un proyecto con sus atributos.

##### Crear actividad

* Si un participante crea una actividad debe tener un título, una descripción, un tipo, una fecha en la que se realizó, una hora de inicio, una hora de finalización y un participante que la realizó.

##### Crear registro de actividad

* Si crea un registro de actividad debe tener por defecto la fecha y hora en la que se inició.

##### Modificar registro de actividad

* Si modifico el registro debe haber un registro de actividad a la cual pueda modificar sus atributos.

##### Crear reporte

* Si creo un reporte debe haber al menos una actividad y debe de contener los atributos tiempo total invertido, tiempo promedio por tipo de actividad, tiempo por cada día, etc.

##### Mostrar reporte de actividades

* Si se va a ver el reporte debe de existir un reporte para poder mostrar.

##### Realizar una colaboración de participantes en una actividad

* Si se hará una colaboración entre participantes, cada participante debe ser capaz de modificar el registro de una actividad.

##### Cronometrar actividad

* Si se empieza una actividad, automáticamente el sistema cronometra el tiempo transcurrido.

##### Pausar cronómetro de actividad

* Si voy a pausar una actividad, manualmente se debe detener el cronómetro.

##### Reanudar cronómetro de actividad

* Si voy a reanudar una actividad, el cronómetro debe estar pausado y manualmente se debe reanudar el cronómetro.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Responsabilidades | Information holder | Structurer | Service provider | Coordinator | Controller | Interfacer |
| 1 |  | X |  |  |  |  |
| 2 |  |  |  | X |  |  |
| 3 | X | X |  |  |  |  |
| 4 | X | X |  |  |  |  |
| 5 |  |  |  |  | X |  |
| 6 | X | X |  |  |  |  |
| 7 | X |  |  |  |  |  |
| 8 |  |  |  | X |  |  |
| 9 |  |  | X |  |  |  |
| 10 |  |  |  |  | X |  |
| 11 |  |  |  |  | X |  |

#### Colaboraciones

Responsabilidades totales:

1. Crear proyecto
2. Agregar participantes
3. Crear actividad
4. Crear registro de actividad
5. Modificar registro de actividad
6. Crear reporte
7. Mostrar reporte de actividades
8. Realizar una colaboración de participantes en una actividad
9. Cronometrar actividad .
10. Pausar cronómetro de actividad
11. Reanudar cronómetro de actividad

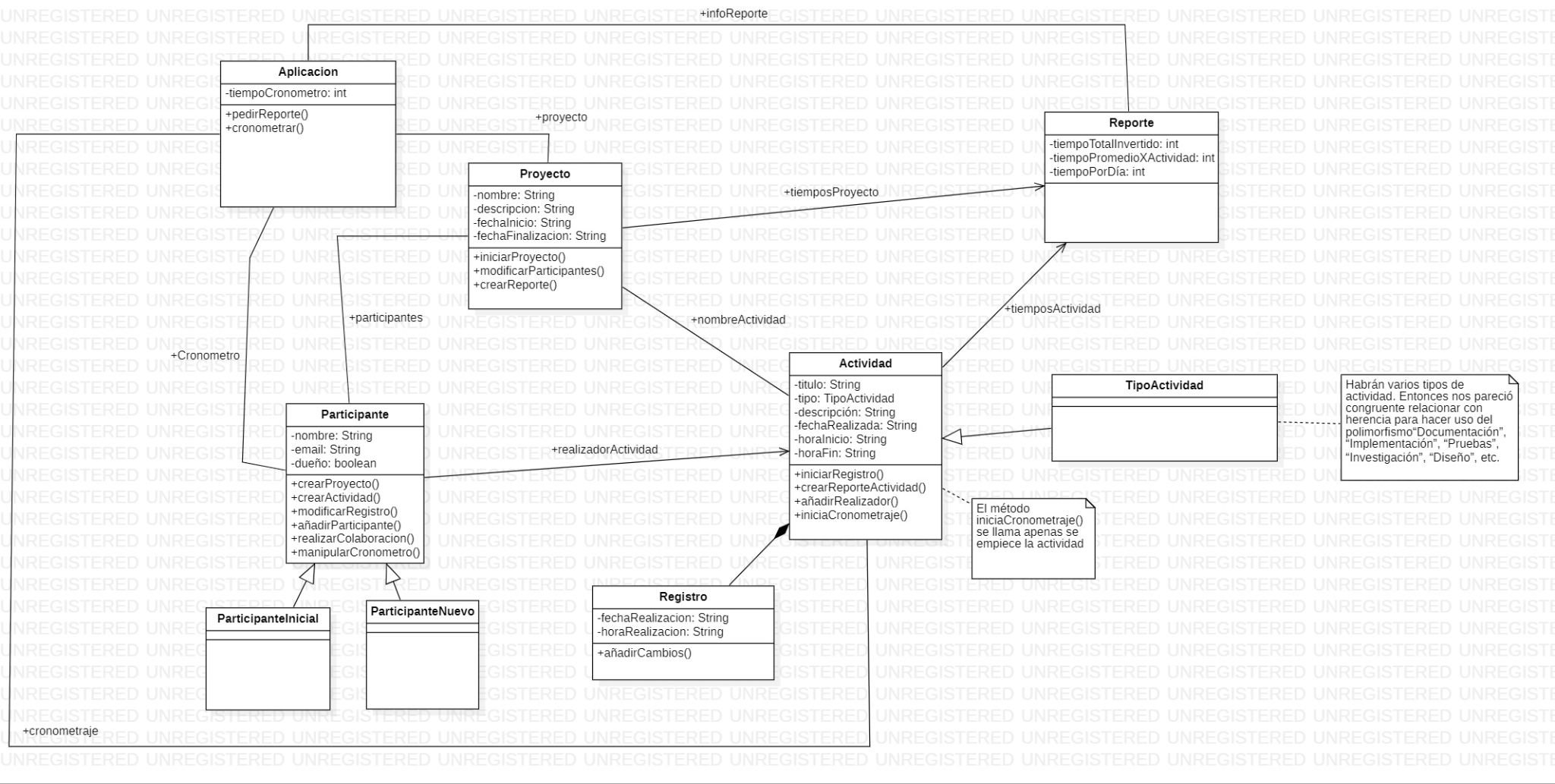
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Responsabilidades | Proyecto | Participante | Actividad | Registro | Reporte | Aplicación |
| 1 | X |  |  |  |  | X |
| 2 | X | X |  |  |  |  |
| 3 |  | X | X |  |  |  |
| 4 |  |  | X | X |  |  |
| 5 |  | X |  | X |  |  |
| 6 | X |  |  |  | X | X |
| 7 | X |  | X |  | X | X |
| 8 |  | X | X |  |  |  |
| 9 |  |  | X |  |  | X |
| 10 |  | X | X |  |  | X |
| 11 |  | X | X |  |  | X |

* Proyecto y Aplicación colaboran porque Aplicación es necesaria para que existan proyectos.
* Proyecto y participante se agrupan con agregar participantes dado a que para agregar participantes debe existir ya un proyecto previamente y un participante para agregar.
* Actividad y participante se agrupan con registrar actividad ya que existe una relación directa porque no puede registrar una actividad si no existe una actividad y además es una función del participante.
* Registro y participante se agrupan con modificar registro ya que existe una relación directa porque no se puede modificar un registro si no existe un registro y porque es una función del participante.
* Actividad y registro se agrupan con crear registro de actividad porque tienen una relación directa ya que para poder realizar la función mencionada anteriormente debe de haber una actividad y un registro.
* Proyecto, reporte y aplicación se agrupan con crear reporte porque para crear un reporte debe de existir un proyecto además tiene una relación directa con reporte y la aplicación es la encargada de realizar dicha función.
* Proyecto, actividad, reporte y aplicación se agrupan con mostrar reporte de actividades porque hay una relación de dependencia dado a que debe de existir un proyecto hará tener una actividad y un reporte para mostrar un reporte además la aplicación en la encargada de desempeñar dicha función.
* Participante y actividad se agrupan con realizar una colaboración de participantes en una actividad porque existe una relación de dependencia ya que para poder realizar la función anteriormente nombrada debe existir previamente un participante y una actividad.
* Actividad y aplicación se agrupan con cronometrar la actividad porque existe una relación directa ya que debe existir una actividad para poderla cronometrar y la aplicación es la encargada de hacer la función anteriormente nombrada.
* Participante, actividad y aplicación se agrupan con pausar cronómetro de actividad y reanudar cronómetro de actividad dado a que existe una relación directa.

#### Diagrama de clases de alto nivel

#### 

#### Diagrama de clases



#### Diagrama de secuencias

#### 

Nota: En el diagrama de secuencias colocamos una consola teniendo en cuenta que es la forma en que los usuarios interactúan con la aplicación.

Nota: En el documento no se pueden apreciar bien las imágenes. Por lo tanto, dejaremos adjunta una [carpeta de drive](https://drive.google.com/drive/folders/1u8g7UYfON3lDT1FSZgMf68jaomC7RfTM?usp=sharing) para que se puedan apreciar mejor.

### Reflexión

En cuanto a ventajas, tenemos que los roles propuestos han logrado obtener el más bajo nivel. Logrando hacer un diagrama de clases. Esto nos permite tener mucha más claridad al momento de implementar el diseño propuesto. Hablando de las desventajas, tenemos que las responsabilidades son difíciles de identificar dado a que al existir pocos roles se tienden a confundir las responsabilidades que le corresponden a cada rol. Además, específicamente el elemento de Tiempo tendría que guardar a medida que van modificando la actividad los jugadores el tiempo que cada uno se demora y al mismo tiempo ir actualizándose. El proceso de diseño fue retador. Hay que pensar en muchas cosas y mientras se lleva a cabo el proceso, es fácil percatarse de cosas que hicieron falta en las iteraciones. Dándose cuenta cada vez de como se podría mejorar el diseño en cada iteración. Lo más problemático fue asignar las responsabilidades y ver cómo colaboran los diferentes roles para cumplirlas. En las siguientes experiencias de diseño, será más fácil tratar de prevenir este mismo problema debido a que sabemos bien que esto es lo más difícil de elaborar en el momento de hacer el diseño y para evitar este problema aplicaremos todo el conocimiento adquirido en la elaboración de este entregable.

#### Trade-Off

En la segunda iteración al asignar las responsabilidades a los roles se presentó el caso en el que a una clase le corresponden varias responsabilidades. Por lo que se puede evidenciar que las responsabilidades aumentaban, pero los roles disminuían. Por otro lado, se puede concluir que hay un estilo de control centralizado esto trae un factor a favor y es que a las responsabilidades estar enfocadas en tres roles clave se agiliza el proceso de tomas de decisiones a la hora de elaborar las colaboraciones