## Tercera Iteración (Final)

### Cambios añadidos:

#### Roles

Los Roles finales quedaron de la siguiente manera:

* **Actividad**: este rol guarda la información de un registro de actividad con sus características: título, descripción, *tipo*, fecha en la que se realizó, hora de inicio, una hora de finalización y *participante* que la realizó.
* **Participante**: en este rol se guarda la información de un participante con las características : nombre y email.
* **Tipo de Actividad**: este rol almacena la clasificación de los tipos de actividad especificados durante la creación del *proyecto*.
* **Proyecto**: Almacena toda la información agrupada de un proyecto: *participantes*, *tipo de actividades, actividades*. Además de sus propias características: nombre, descripción, fecha de inicio y fecha de finalización. Es importante destacar que dentro de estas características se debe asociar un participante propietario del proyecto.
* **Cronómetro**: este rol es el encargado de crear los reportes de tiempo de una *actividad* en específico.

Nota importante: en la implementación se creó una clase llamada “Aplicación”, la cual se encarga de conectar las funciones entre la consola y los roles mencionados anteriormente.

#### 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 que para poder mostrar un reporte es necesario que antes de eso, se cree un reporte.
2. Agregar participantes a un proyecto
   * Para tener registrado cuales son los participantes de un proyecto es necesario agregarlos dentro de un proyecto.
3. Crear proyecto:
   * La mayor responsabilidad. Al crear un proyecto se le asocia un grupo de actividades, un grupo de participantes, un grupo de los tipos de actividades que se desempeñarán, y sus características propias.

#### 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 |
| --- | --- | --- | --- | --- | --- | --- |
| Actividad |  | X |  |  |  |  |
| Participante |  |  |  |  | X |  |
| TipoActividad | X |  |  |  |  |  |
| Proyecto |  | X |  | X |  |  |
| Cronómetro |  |  | X |  |  |  |

### Responsabilidades

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

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

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 tendrá 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

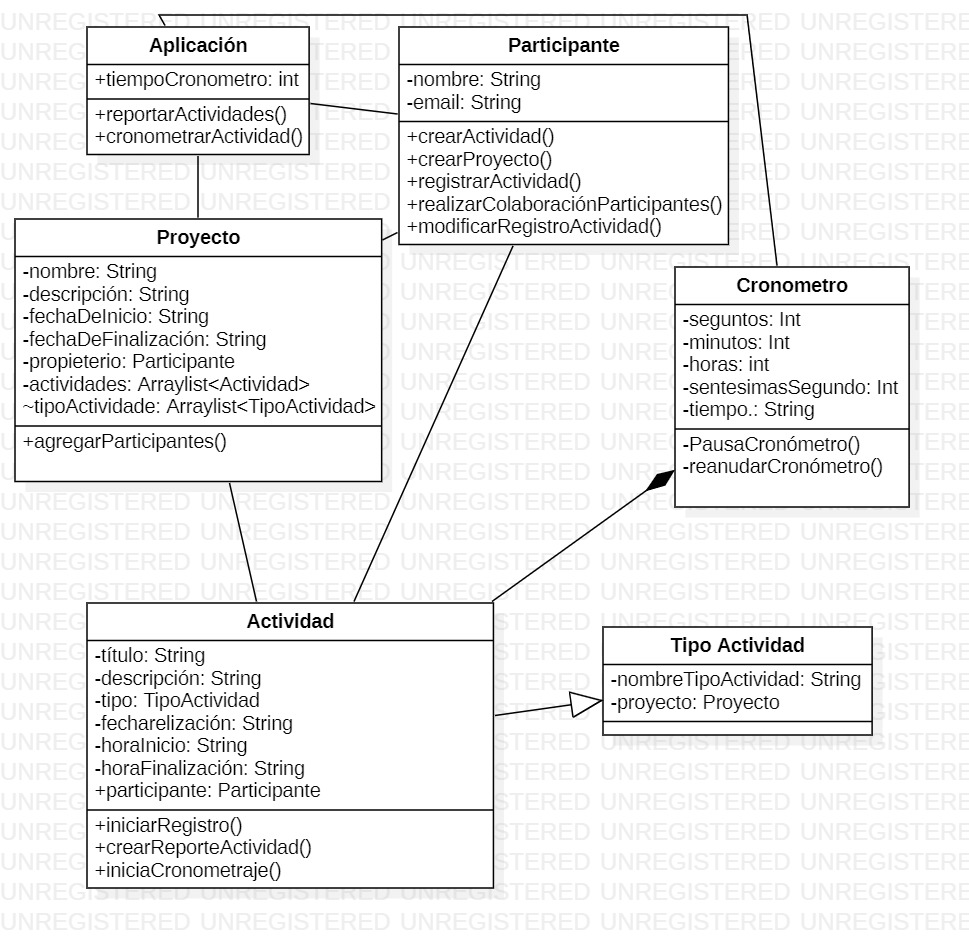
| Responsabilidades | Proyecto | Participante | Actividad | Cronómetro | Tipo Actividad | Aplicación |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | X | X |  |  |  |  |
| 2 | X | X |  |  |  |  |
| 3 |  | X | X |  | X |  |
| 4 |  | X | X |  |  | X |
| 5 |  | X |  |  |  |  |
| 6 |  |  | x |  |  | X |
| 7 | X |  | X |  |  | X |
| 8 |  | X | X |  | X |  |
| 9 |  |  | X |  |  | x |
| 10 |  |  |  | x |  |  |
| 11 |  |  |  | x |  |  |

* Proyecto y participante colaboran porque el participante es quien tiene la función de crear un proyecto
* 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.
* Participante,actividad y tipo de actividad se agrupan cuando se crea una actividad ya que el participante es quien crea la actividad y al crearla debe establecer qué tipo de actividad es.
* 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.
* Actividad 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 aplicaciones se agrupan para crear un reporte ya que debe existir una actividad para crear un reporte y además quien lo hace es la aplicación
* Proyecto, actividad y aplicación se agrupan con mostrar reporte de actividades porque hay una relación de dependencia dado a que debe de existir un proyecto para tener una actividad y para mostrar un reporte además la aplicación es la encargada de desempeñar dicha función.
* Participante,actividad y tipo de 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 para agregar ,una actividad y al agregar un participante se debe tener claro qué tipo de actividad desea agregarlo .
* 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.

#### 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.

### 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 cómo 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

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 las responsabilidades estar enfocadas en tres roles clave se agiliza el proceso de tomas de decisiones a la hora de elaborar las colaboraciones

## 