Taller de Pensamiento Crítico con CU (entrevista) y con CU (PlantUML)

Tema: Relación entre el Análisis de Requisitos Funcionales y la Generación de Casos de Uso con IAGen (PlantUML)

Asignatura: Análisis y Diseño de Software

Duración: 40 minutos

Objetivo del Taller: Desarrollar el pensamiento crítico de los estudiantes mediante la comparación entre los Casos de Uso derivados del proceso de análisis de requisitos funcionales de su ERS (SRS) y los Casos de Uso modelados gráficamente mediante PlantUML, evaluando la coherencia, trazabilidad y completitud de los modelos.

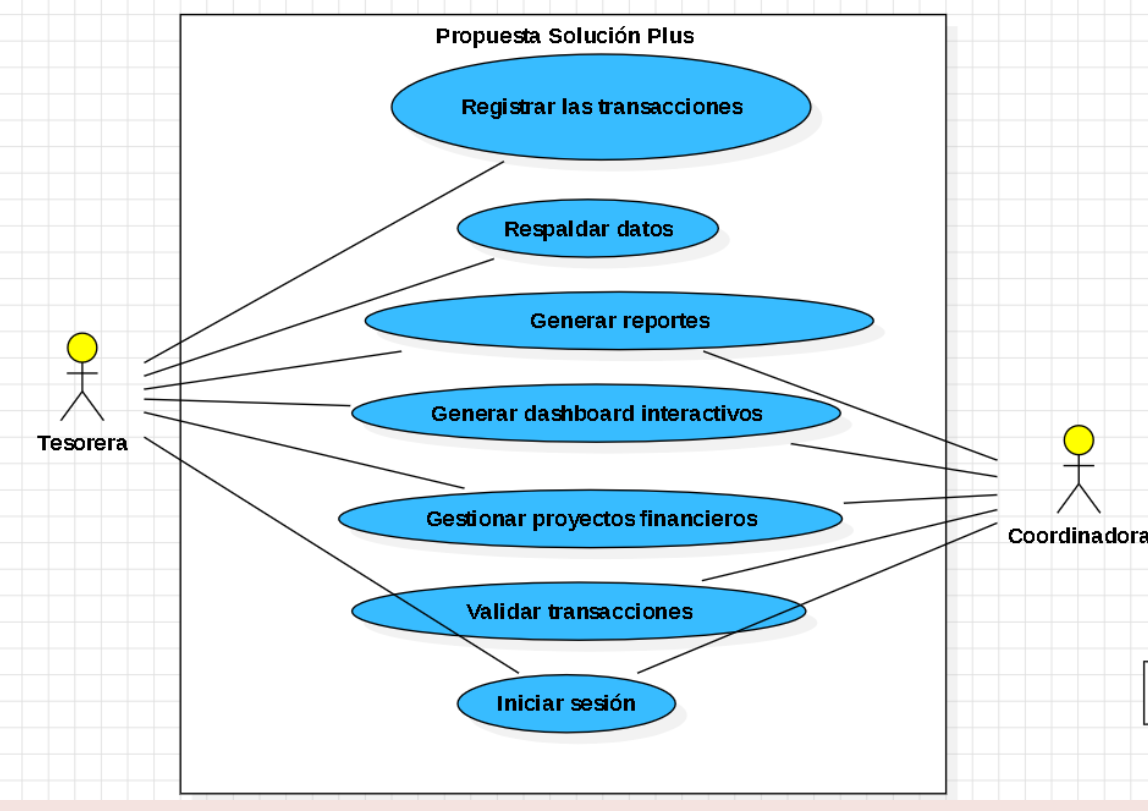
## Contexto de Aprendizaje

Durante el proceso de análisis de requisitos, los analistas obtienen los requisitos funcionales (RF) a partir de entrevistas, observaciones y documentación. Cada requisito funcional puede representarse mediante un caso de uso, el cual describe cómo un actor interactúa con el sistema para alcanzar un objetivo específico. En esta práctica, los estudiantes contrastarán cómo los requisitos funcionales textuales se transforman en diagramas de casos de uso generados con PlantUML.

## Actividades

### Parte 1. Lectura y análisis (10 min)

**Caso de Uso 1**

****

@startuml

left to right direction

actor Tesorera

actor Coordinadora

rectangle "Propuesta Solución Plus" {

usecase "Registrar las transacciones" as CU1

usecase "Respaldar datos" as CU2

usecase "Generar reportes" as CU3

usecase "Generar dashboard interactivos" as CU4

usecase "Gestionar proyectos financieros" as CU5

usecase "Validar transacciones" as CU6

usecase "Iniciar sesión" as CU7

}

' Relaciones de Tesorera

Tesorera --> CU1

Tesorera --> CU2

Tesorera --> CU3

Tesorera --> CU4

Tesorera --> CU5

Tesorera --> CU6

Tesorera --> CU7

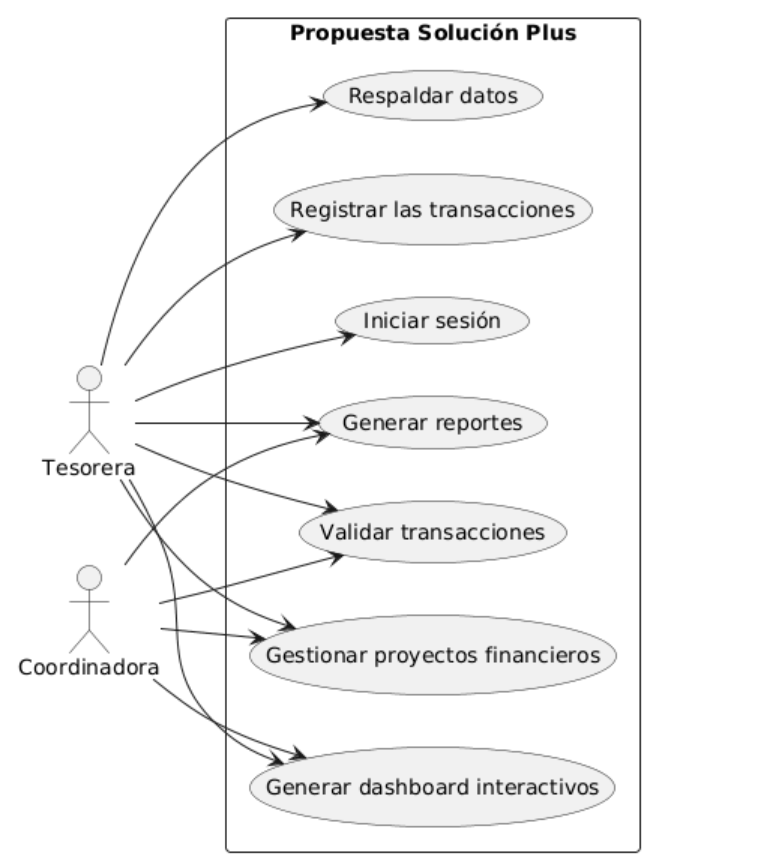
' Relaciones de Coordinadora

Coordinadora --> CU3

Coordinadora --> CU4

Coordinadora --> CU5

Coordinadora --> CU6

@enduml

**Hay dos actores principales:**

* **Actor** **Tesorera:** Tiene relación con la mayoría de las funcionalidades (7 de 8 casos de uso).
* **Actor** **Coordinadora**: Tiene relación con un subconjunto de funcionalidades, principalmente de consulta y gestión (3 de 8 casos de uso).

**Acciones Principales**

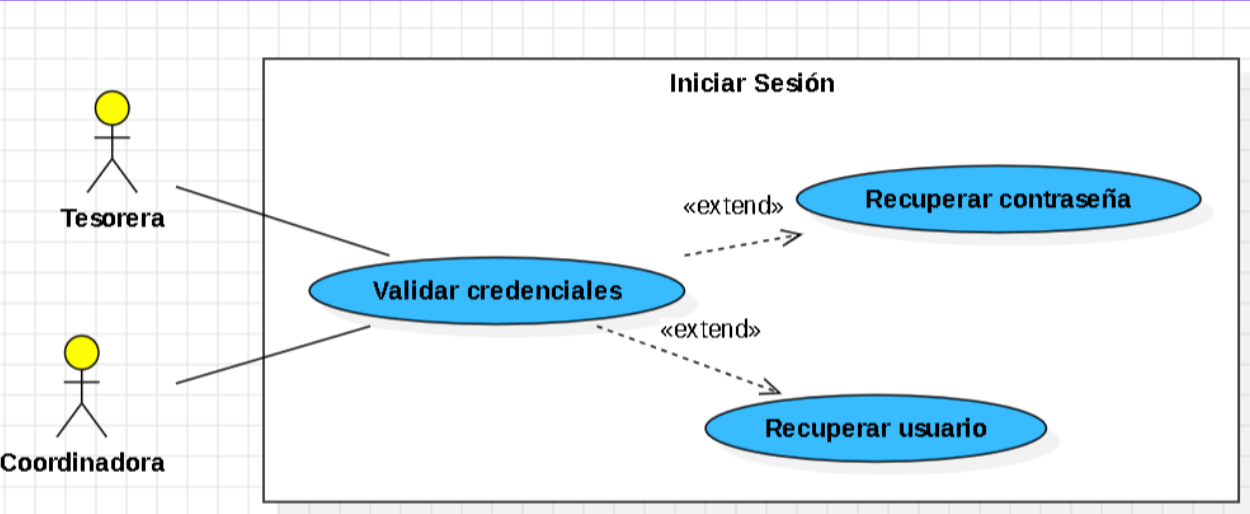
Estas son las acciones principales que el sistema debe ejecutar, tal como se presentan en el diagrama:

1. Registrar las transacciones
2. Respaldar datos
3. Generar reportes
4. Generar dashboard interactivos
5. Gestionar proyectos financieros
6. Validar transacciones
7. Iniciar sesión

**Relaciones de Inclusión o Extensión**

En este diagrama de Casos de Uso no aparece ninguna relación de inclusión (<<include>>) ni de extensión (<<extend>>).

**Caso de Uso 2**



@startuml

left to right direction

actor Tesorera

actor Coordinadora

rectangle "Iniciar Sesión" {

usecase "Validar credenciales" as UC\_Validar

usecase "Recuperar contraseña" as UC\_RecuperarContrasena

usecase "Recuperar usuario" as UC\_RecuperarUsuario

UC\_Validar <.left. UC\_RecuperarContrasena : <<extend>>

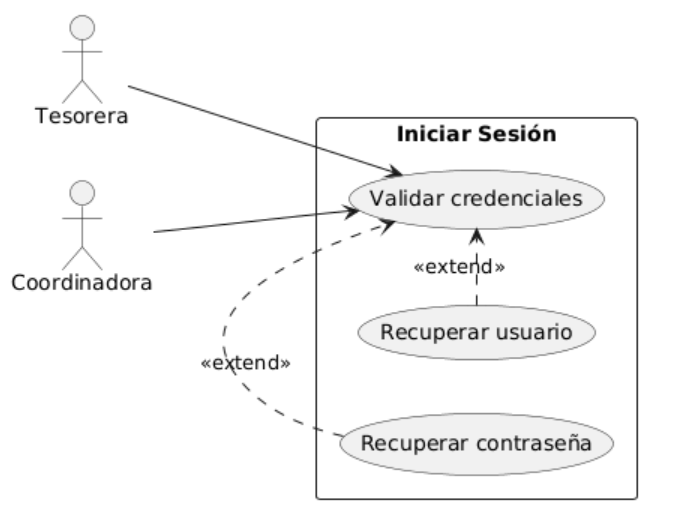
UC\_Validar <.left. UC\_RecuperarUsuario : <<extend>>

}

Tesorera --> UC\_Validar

Coordinadora --> UC\_Validar

@enduml

****

**Hay dos actores principales:**

Ambos actores, la **Tesorera** y la **Coordinadora**, son actores principales en este requisito, ya que ambos inician directamente el caso de uso central: **Validar credenciales**.

**Acciones Principales**

Las acciones principales del sistema son:

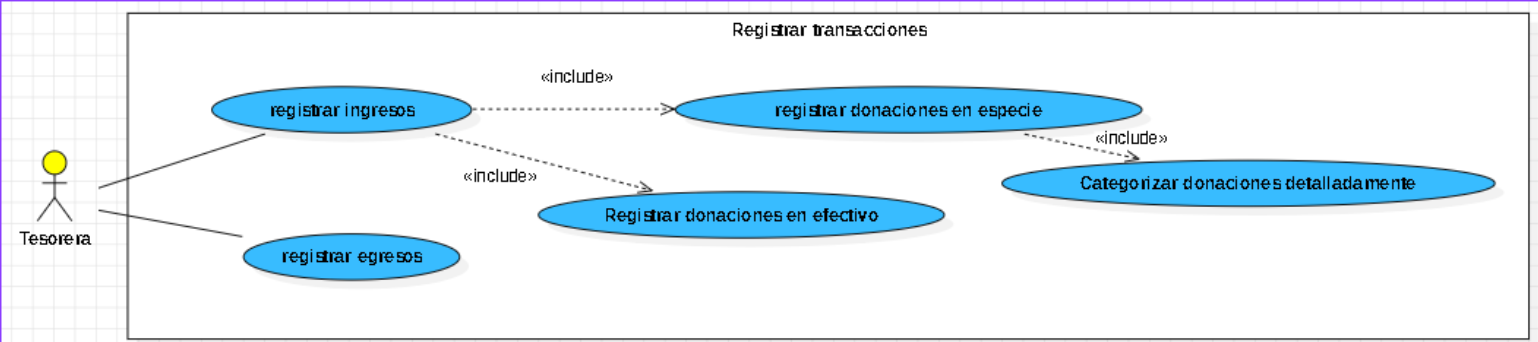
1. **Validar credenciales:** El caso de uso base para permitir el acceso al sistema.
2. **Recuperar contraseña:** Una acción opcional para el usuario que intenta iniciar sesión.
3. **Recuperar usuario:** Otra acción opcional para el usuario que intenta iniciar sesión.

**Relaciones de Inclusión o Extensión**

En este diagrama se utilizan exclusivamente relaciones de **Extensión** (<<extend>>):

* **<<extend>>** de **Recuperar contraseña** a **Validar credenciales**.
* **<<extend>>** de **Recuperar usuario** a **Validar credenciales**.

**Caso de Uso 3**



@startuml

left to right direction

actor Tesorera

rectangle "Registrar transacciones" {

usecase "registrar ingresos" as UC\_Ingresos

usecase "registrar egresos" as UC\_Egresos

usecase "registrar donaciones en especie" as UC\_Especie

usecase "Registrar donaciones en efectivo" as UC\_Efectivo

usecase "Categorizar donaciones detalladamente" as UC\_Categorizar

UC\_Ingresos .right.> UC\_Especie : <<include>>

UC\_Ingresos .down.> UC\_Efectivo : <<include>>

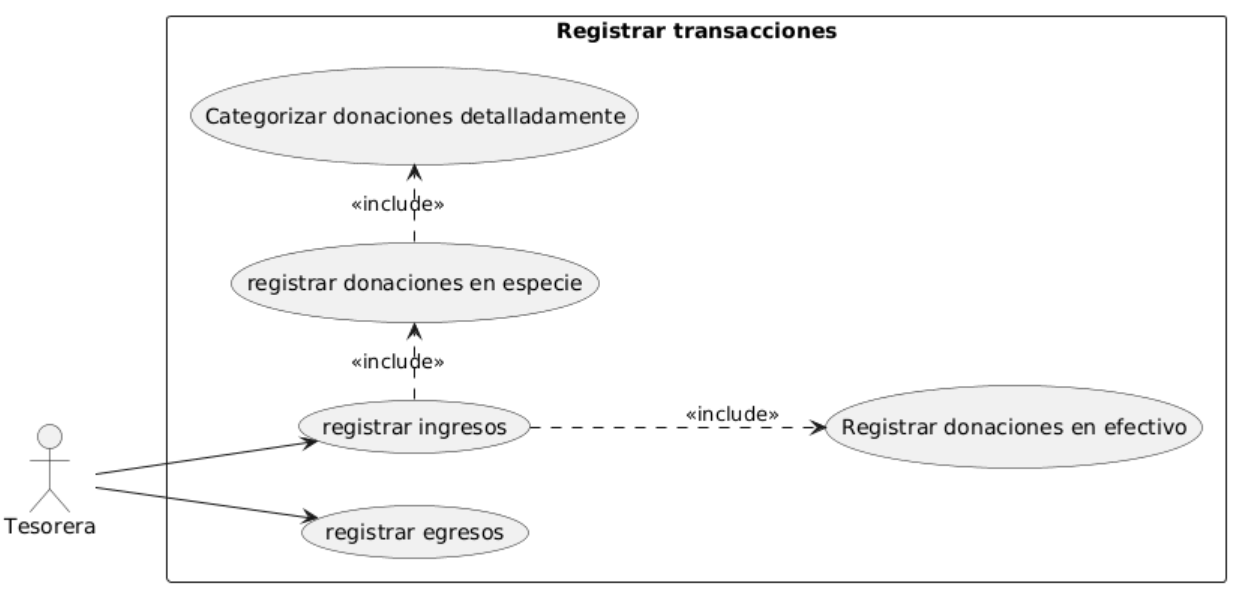
UC\_Especie .right.> UC\_Categorizar : <<include>>

}

Tesorera --> UC\_Ingresos

Tesorera --> UC\_Egresos

@enduml



**Hay un actor principal:**

* **Tesorera**: Es el único actor que interactúa con este grupo de casos de uso, iniciando las acciones de registro.

**Acciones Principales**

Las acciones principales que el sistema debe ejecutar son:

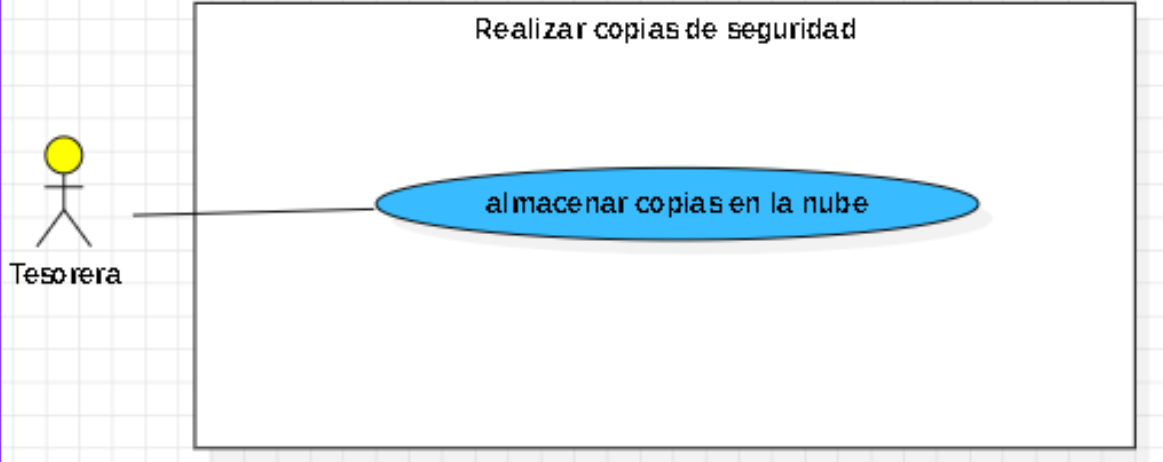
1. registrar ingresos (Caso Base).
2. registrar egresos (Caso Base).
3. registrar donaciones en especie (Caso Incluido por registrar ingresos).
4. Registrar donaciones en efectivo (Caso Incluido por registrar ingresos).
5. Categorizar donaciones detalladamente (Caso Incluido por registrar donaciones en especie).

**Relaciones de Inclusión o Extensión**

Este diagrama utiliza exclusivamente relaciones de <<include>>, las cuales indican que el comportamiento del caso incluido es obligatorio dentro del caso base:

1. **<<include>>** de **registrar ingresos** a **registrar donaciones en especie**.
2. **<<include>>** de **registrar ingresos** a **Registrar donaciones en efectivo**.
3. **<<include>>** de **registrar donaciones en especie** a **Categorizar donaciones detalladamente**.

**Caso de Uso 4**



@startuml

left to right direction

actor Tesorera

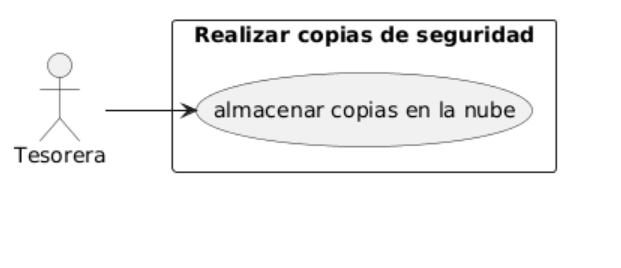
rectangle "Realizar copias de seguridad" {

usecase "almacenar copias en la nube" as UC\_Almacenar

}

Tesorera --> UC\_Almacenar

@enduml



**Hay un actor principal:**

* **Tesorera**: Es el único actor que interactúa con este caso de uso.

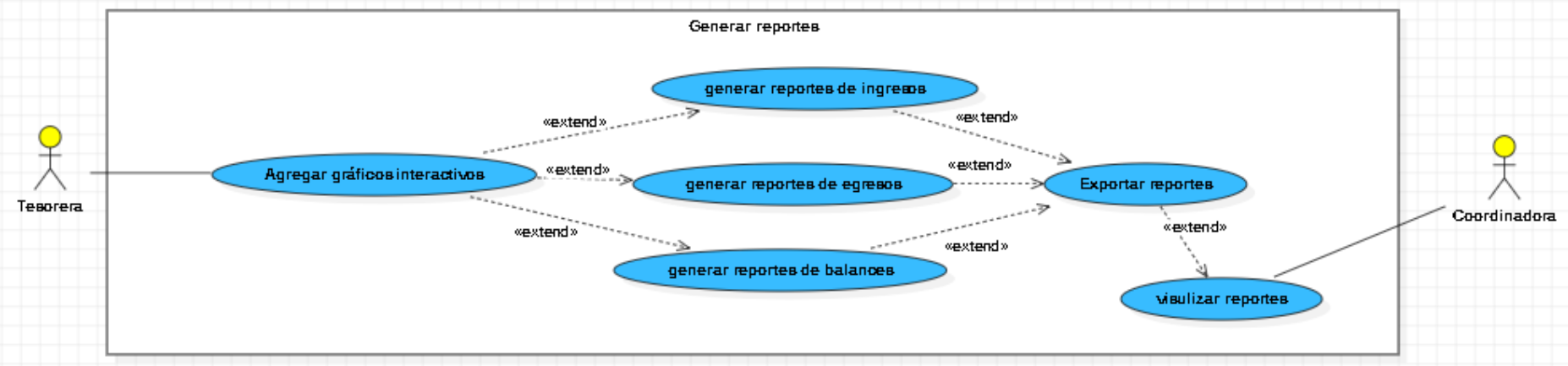
**Acciones Principales**

1. almacenar copias en la nube

**Relaciones de Inclusión o Extensión**

En este diagrama no aparece ninguna relación de inclusión <<include>>)ni de extensión <<extend>>.

**Caso de Uso 5**



@startuml

left to right direction

actor Tesorera

actor Coordinadora

rectangle "Generar reportes" {

usecase "Agregar gráficos interactivos" as UC\_Graficos

usecase "generar reportes de ingresos" as UC\_Ingresos

usecase "generar reportes de egresos" as UC\_Egresos

usecase "generar reportes de balances" as UC\_Balances

usecase "Exportar reportes" as UC\_Exportar

usecase "visualizar reportes" as UC\_Visualizar

' Extensiones de Agregar gráficos interactivos

UC\_Graficos <.left. UC\_Ingresos : <<extend>>

UC\_Graficos <.left. UC\_Egresos : <<extend>>

UC\_Graficos <.left. UC\_Balances : <<extend>>

' Extensiones de Exportar reportes

UC\_Ingresos <.right. UC\_Exportar : <<extend>>

UC\_Egresos <.right. UC\_Exportar : <<extend>>

UC\_Balances <.right. UC\_Exportar : <<extend>>

' Extensión de Visualizar reportes

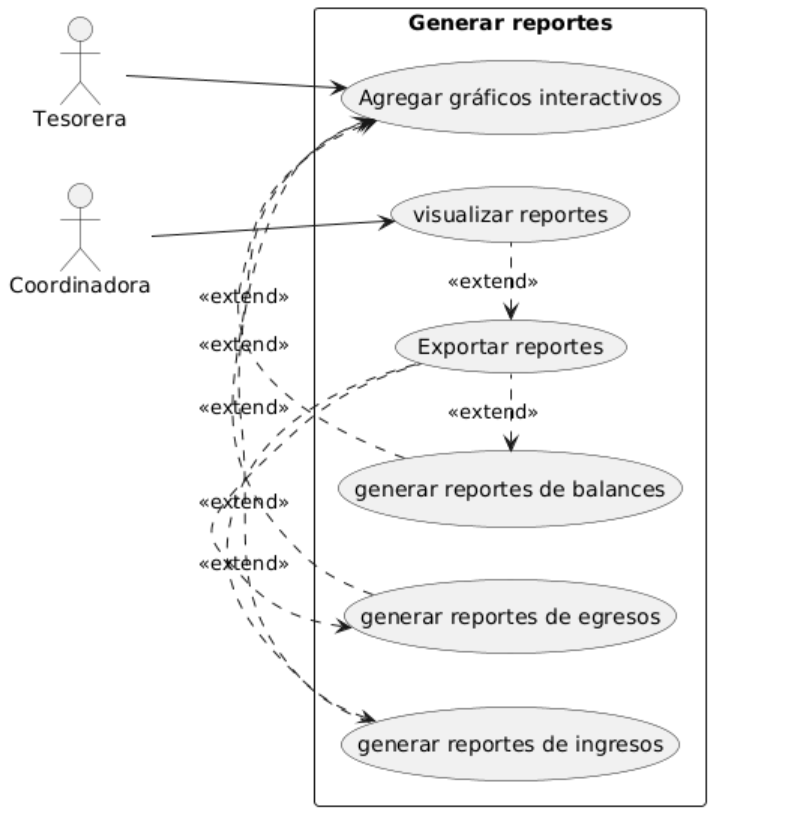
UC\_Exportar <.right. UC\_Visualizar : <<extend>>

}

Tesorera --> UC\_Graficos

Coordinadora --> UC\_Visualizar

@enduml



**Hay dos actores:**

**Actor Principal (Tesorera):** Inicia la acción principal de Agregar gráficos interactivos.

**Actor Secundario (Coordinadora):** Inicia la acción de visualizar reportes (aunque esta acción también es una extensión de Exportar reportes).

**Acciones Principales**

Las acciones principales que el sistema debe ejecutar son:

1. Agregar gráficos interactivos (Caso Base iniciado por la Tesorera).
2. generar reportes de ingresos (Extensión de Agregar gráficos interactivos).
3. generar reportes de egresos (Extensión de Agregar gráficos interactivos).
4. generar reportes de balances (Extensión de Agregar gráficos interactivos).
5. Exportar reportes (Extensión de los 3 casos de generación de reportes).
6. visualizar reportes (Extensión de Exportar reportes e iniciada por la Coordinadora).

**Relaciones de Inclusión o Extensión**

Este diagrama utiliza exclusivamente relaciones de Extensión (<<extend>>):

1. **Tres extensiones de Agregar gráficos interactivos:**

De generar reportes de ingresos a Agregar gráficos interactivos.

De generar reportes de egresos a Agregar gráficos interactivos.

De generar reportes de balances a Agregar gráficos interactivos.

1. **Tres extensiones de Exportar reportes:**

De Exportar reportes a generar reportes de ingresos.

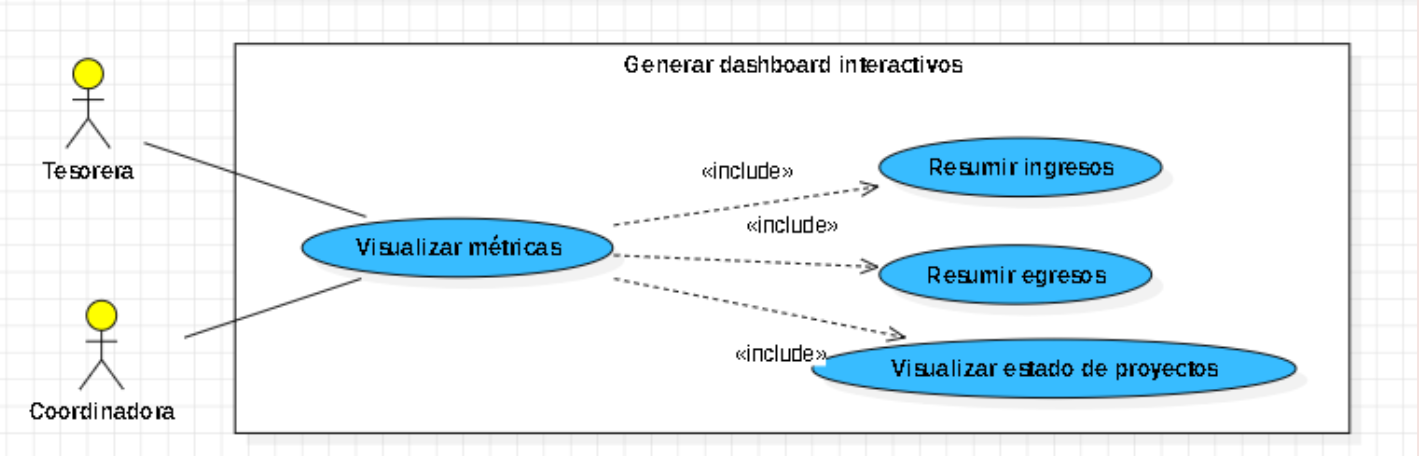
De Exportar reportes a generar reportes de egresos.

De Exportar reportes a generar reportes de balances.

1. **Una extensión de visualizar reportes:**

De visualizar reportes a Exportar reportes.

**Caso de Uso 6**



@startuml

left to right direction

actor Tesorera

actor Coordinadora

rectangle "Generar dashboard interactivos" {

usecase "Visualizar métricas" as UC\_Metricas

usecase "Resumir ingresos" as UC\_Ingresos

usecase "Resumir egresos" as UC\_Egresos

usecase "Visualizar estado de proyectos" as UC\_Proyectos

UC\_Metricas .right.> UC\_Ingresos : <<include>>

UC\_Metricas .right.> UC\_Egresos : <<include>>

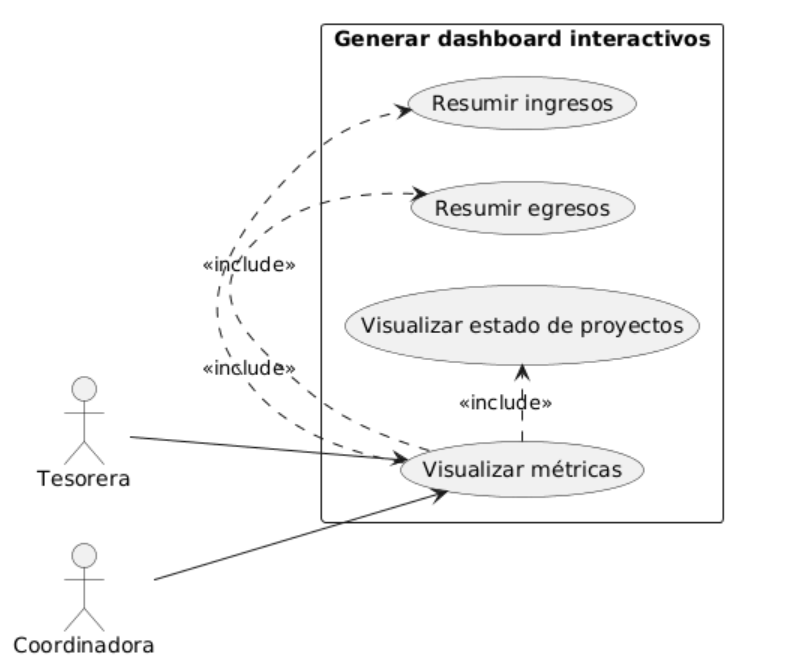
UC\_Metricas .right.> UC\_Proyectos : <<include>>

}

Tesorera --> UC\_Metricas

Coordinadora --> UC\_Metricas

@enduml

****

**Hay dos actores principales:**

Tesorera y Coordinadora: Ambos inician el caso de uso base central de visualizar métricas.

**Acciones Principales**

Las acciones principales que el sistema debe ejecutar son:

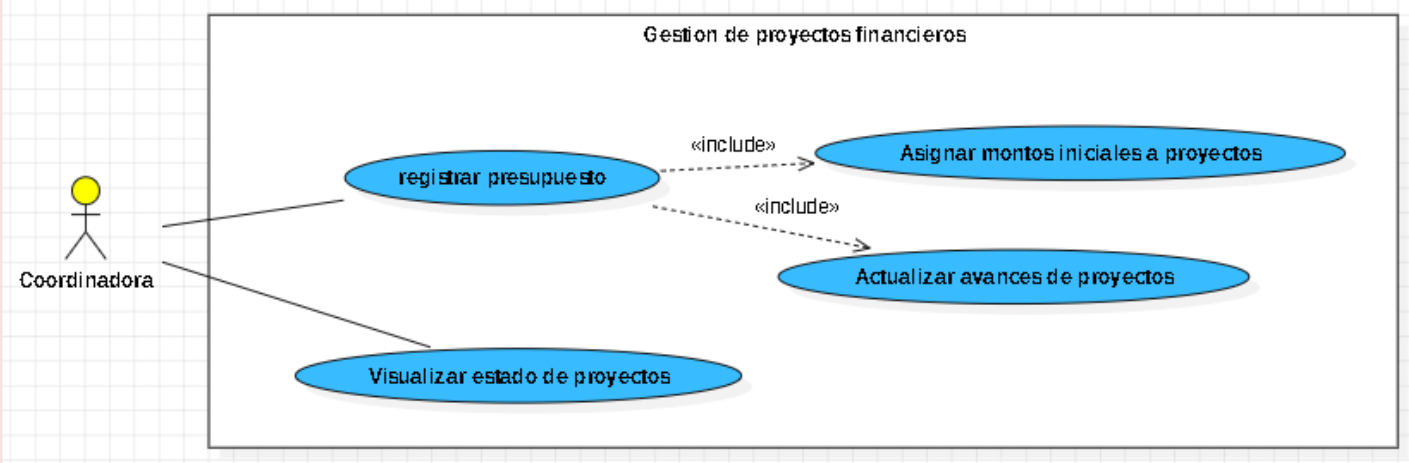
1. Visualizar métricas (Caso Base iniciado por ambos actores).
2. Resumir ingresos (Caso Incluido).
3. Resumir egresos (Caso Incluido).
4. Visualizar estado de proyectos (Caso Incluido).

**Relaciones de Inclusión o Extensión**

Este diagrama utiliza tres relaciones de **Inclusión** (<<include>>):

* <<include>> de Visualizar métricas a Resumir ingresos.
* <<include>> de Visualizar métricas a Resumir egresos.
* <<include>> de Visualizar métricas a Visualizar estado de proyectos.

**Caso de Uso 7**

****

@startuml

left to right direction

actor Coordinadora

rectangle "Gestión de proyectos financieros" {

usecase "registrar presupuesto" as UC\_Presupuesto

usecase "Visualizar estado de proyectos" as UC\_Visualizar

usecase "Asignar montos iniciales a proyectos" as UC\_Asignar

usecase "Actualizar avances de proyectos" as UC\_Actualizar

UC\_Presupuesto .right.> UC\_Asignar : <<include>>

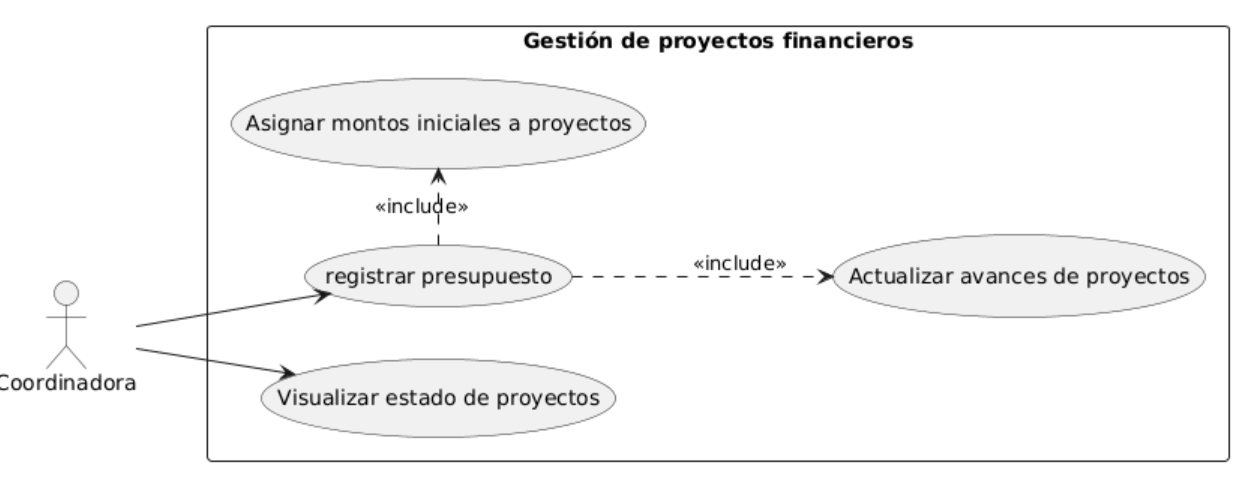
UC\_Presupuesto .down.> UC\_Actualizar : <<include>>

}

Coordinadora --> UC\_Presupuesto

Coordinadora --> UC\_Visualizar

@enduml



**Hay un actor principal:**

Coordinadora: Es la única que interactúa en este contexto de gestión de proyectos.

**Acciones Principales**

Las acciones principales que el sistema debe ejecutar son:

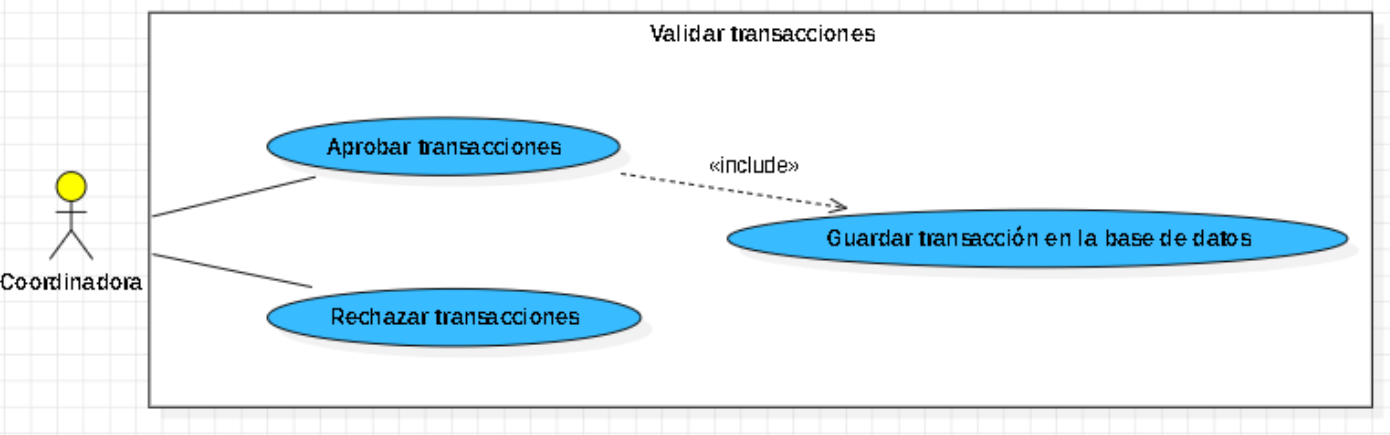
1. registrar presupuesto (Caso Base).
2. Visualizar estado de proyectos (Caso Base independiente).
3. Asignar montos iniciales a proyectos (Caso Incluido).
4. Actualizar avances de proyectos (Caso Incluido).

**Relaciones de Inclusión o Extensión**

Este diagrama utiliza dos relaciones de <<include>>:

* <<include>> de registrar presupuesto a Asignar montos iniciales a proyectos.
* <<include>> de registrar presupuesto a Actualizar avances de proyectos.

**Caso de Uso 8**



@startuml

left to right direction

actor Coordinadora

rectangle "Validar transacciones" {

usecase "Aprobar transacciones" as UC\_Aprobar

usecase "Rechazar transacciones" as UC\_Rechazar

usecase "Guardar transacción en la base de datos" as UC\_Guardar

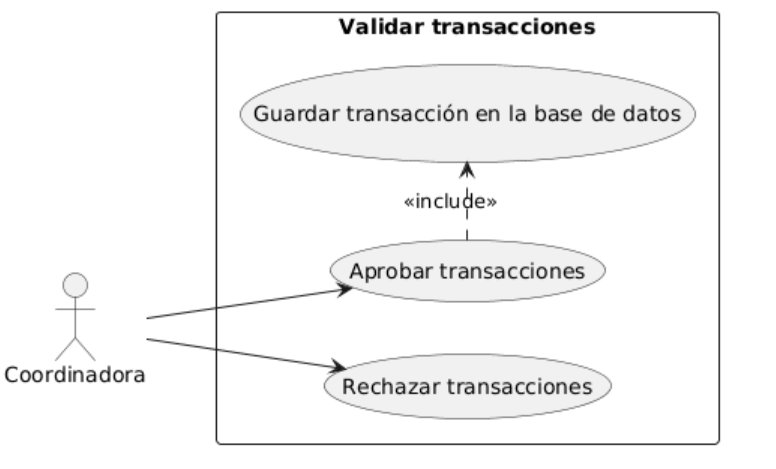
UC\_Aprobar .right.> UC\_Guardar : <<include>>

}

Coordinadora --> UC\_Aprobar

Coordinadora --> UC\_Rechazar

@enduml



**Hay un actor principal**

Coordinadora: Es la única que interactúa en este proceso de validación.

**Acciones Principales**

Las acciones principales que el sistema debe ejecutar son:

1. Aprobar transacciones (Caso Base).
2. Rechazar transacciones (Caso Base).
3. Guardar transacción en la base de datos (Caso Incluido).

**Relaciones de Inclusión o Extensión**

<<include>> de Aprobar transacciones a Guardar transacción en la base de datos.

### Parte 2. Pensamiento crítico (20 min)

Responde con tus propias palabras:

**1. ¿Qué diferencias observas entre los Casos de Uso derivados de entrevistas o descripciones textuales y los Casos de Uso generados automáticamente en PlantUML?**  
Las diferencias fundamentales radican en la Precisión del Modelo y la Claridad Visual entre el modelo generado a partir del análisis humano y el modelo generado mediante código PlantUML si se pueden ver los casos de uso derivados de entrevistas pueden ser un poco mas claro con respecto a los que genera el PlantUM ya que pueden sobreponerse algunas relaciones principalmente de <<extend>> y de <<include>>

2. ¿De qué manera el uso de PlantUML facilita (o limita) el trabajo del analista al modelar los requisitos funcionales?

El uso de PlantUML actúa como un acelerador de la formalización de requisitos, ofreciendo grandes ventajas en la eficiencia y la comunicación, aunque presenta algunas limitaciones iniciales como la curva de aprendizaje inicial, perdida de interacción gráfica directa o detalle fino del modelo ya que estos de alguna manera limita algunas cosas que podemos realizarlo de forma manual