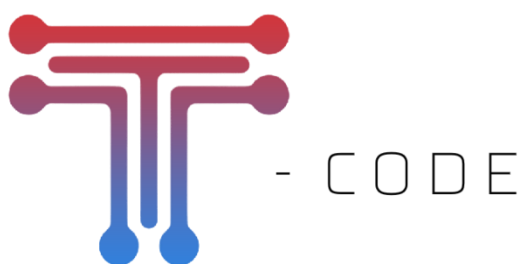


# Modelo de Diseño

Vesta Risk Manager

T-Code

Agustín Collareda, Cintia Hernandez, Hugo Frey



El propósito del Modelo de Diseño es empezar a realizar los casos de uso desarrollados durante las etapas anteriores en especial la etapa de captura de requerimientos. Es decir, tomar el Modelo de Casos de Uso y las Especificaciones Suplementarias creadas con anterioridad entre otros insumos y generar un modelo de diseño que pueda ser usado por los desarrolladores durante la etapa de implementación.



## Tabla de contenido

<b>Introducción .....</b>	<b>5</b>
<i>Propósito .....</i>	<i>5</i>
<i>Alcance .....</i>	<i>5</i>
<b>Referencias.....</b>	<b>5</b>
<i>Visión general.....</i>	<i>6</i>
<b>Diseño de Casos de Uso .....</b>	<b>6</b>
<i>Caso de Uso 1: Autenticarse.....</i>	<i>6</i>
Especificación de caso de uso de diseño .....	6
Diagrama de paquetes.....	6
Diagrama de Interacción.....	7
Requerimientos especiales o de implementación .....	7
<i>Caso de Uso 2: Administrar acceso al sistema .....</i>	<i>7</i>
Especificación de caso de uso de diseño .....	7
Diagrama de paquetes.....	9
Diagrama de Interacción.....	11
<i>Caso de Uso 3: Administrar proyectos .....</i>	<i>12</i>
Especificación de caso de uso de diseño .....	12
Diagrama de paquetes.....	13
Diagrama de Interacción.....	15
<i>Caso de Uso 4: Añadir riesgo a la lista .....</i>	<i>16</i>
Especificación de caso de uso de diseño .....	16
Diagrama de paquetes.....	16
Diagrama de Interacción.....	18
<i>Caso de Uso 5: Modificar lista de riesgos.....</i>	<i>18</i>
Especificación de caso de uso de diseño .....	18
Diagrama de paquetes.....	19
Diagrama de Interacción.....	21
<i>Caso de Uso 6: Administrar categorías de riesgos .....</i>	<i>22</i>
Especificación de caso de uso de diseño .....	22
Diagrama de paquetes.....	22
Diagrama de Interacción.....	23
<i>Caso de Uso 7: Realizar evaluación de riesgo .....</i>	<i>24</i>
Especificación de caso de uso de diseño .....	24
Diagrama de paquetes.....	24
Diagrama de Interacción.....	26
<i>Caso de Uso 8: Añadir plan de riesgo .....</i>	<i>26</i>
Especificación de caso de uso de diseño .....	26
Diagrama de paquetes.....	27

Diagrama de Interacción.....	29
<i>Caso de Uso 9: Modificar plan de acción.....</i>	<i>29</i>
Especificación de caso de uso de diseño .....	29
Diagrama de paquetes.....	30
Diagrama de Interacción.....	32
<i>Caso de Uso 10: Realizar informes .....</i>	<i>32</i>
Especificación de caso de uso de diseño .....	32
Diagrama de paquetes.....	33
Diagrama de Interacción.....	35
<i>Caso de Uso 11: Exportar archivos .....</i>	<i>36</i>
Especificación de caso de uso de diseño .....	36
Diagrama de paquetes.....	36
Diagrama de Interacción.....	37
<i>Caso de Uso 12: Realizar análisis de riesgo .....</i>	<i>37</i>
Especificación de caso de uso de diseño .....	37
Diagrama de paquetes.....	38
Diagrama de Interacción.....	40
<b>Diagramas .....</b>	<b>41</b>
<i>Diagrama de componentes .....</i>	<i>41</i>
<i>Diagrama de Clases.....</i>	<i>42</i>
<i>Diagramas de Paquetes .....</i>	<i>43</i>

# Modelo de Diseño

---

## Introducción

El presente documento detalla el modelo de diseño para nuestro proyecto, Vesta Risk Manager. Este documento es una pieza clave en el proceso de desarrollo, ya que proporciona una visión integral de la arquitectura, componentes y estructura general del sistema que estamos construyendo. Además, este documento actúa como una guía invaluable durante la fase de implementación. Proporciona a los desarrolladores una hoja de ruta clara, reduciendo la posibilidad de malentendidos y errores en la construcción del sistema.

## Propósito

El propósito principal de este documento es proporcionar una visión clara y completa del diseño de Vesta Risk Manager. Sirve como un punto de referencia para todos los miembros del equipo de desarrollo, diseñadores, y stakeholders involucrados en el proyecto.

## Alcance

Este documento de Modelo de Diseño abarca la totalidad de Vesta Risk Manager, incluyendo todos sus componentes, módulos y funcionalidades. El alcance se extiende a la arquitectura general del sistema, el diseño de la interfaz de usuario, la estructura de la base de datos, los componentes del back-end, las integraciones con sistemas externos, los flujos de datos y procesos principales, así como las consideraciones de seguridad y rendimiento.

## Referencias

Los documentos a los que se le hace referencia en este documento son:

- Propuesta de desarrollo.
- Especificación de requerimientos.
- Modelo de casos de uso.
- Modelo de datos.
- Arquitectura del sistema.

Estos documentos fueron realizados por el equipo de T-Code.

## Visión general

## Diseño de Casos de Uso

### Caso de Uso 1: Autenticarse

#### Especificación de caso de uso de diseño

CASO DE USO N°1	Autenticarse	
Actor Principal	Administrador del sistema; Líder del proyecto; Desarrollador	
Precondiciones	El actor ingresa a la página index del sistema	
Disparador	El actor selecciona “iniciar sesión” en la página de index	
DESCRIPCION	Paso	Acción
	1	El actor inicia sesión con su cuenta de Google.
	2	La <u>InterfazUsuario</u> solicita <i>iniciarSesion(googleUser)</i> al <u>GestorUsuario</u> .
	3	El <u>GestorUsuario</u> envía el evento <i>verificarUsuario(email)</i> a <u>Usuario</u> , <u>Usuario</u> solicita <i>verificarUsuario(email)</i> a la <u>BDD</u> , la <u>BDD</u> verifica el usuario, la <u>BDD</u> retorna la respuesta a <u>Usuario</u> , <u>Usuario</u> retorna la respuesta al <u>GestorUsuario</u> (3a). El <u>GestorUsuario</u> retorna la respuesta a la <u>InterfazUsuario</u> . La <u>InterfazUsuario</u> redirige al actor a la página de inicio del sistema.
EXTENSIONES	Paso	Acción de Derivación
	3a	El actor no se encuentra registrado en la base de datos. El <u>GestorUsuario</u> retorna la respuesta a la <u>InterfazUsuario</u> . La <u>interfazUsuario</u> redirige al actor a la página index del sistema y muestra un mensaje de error.

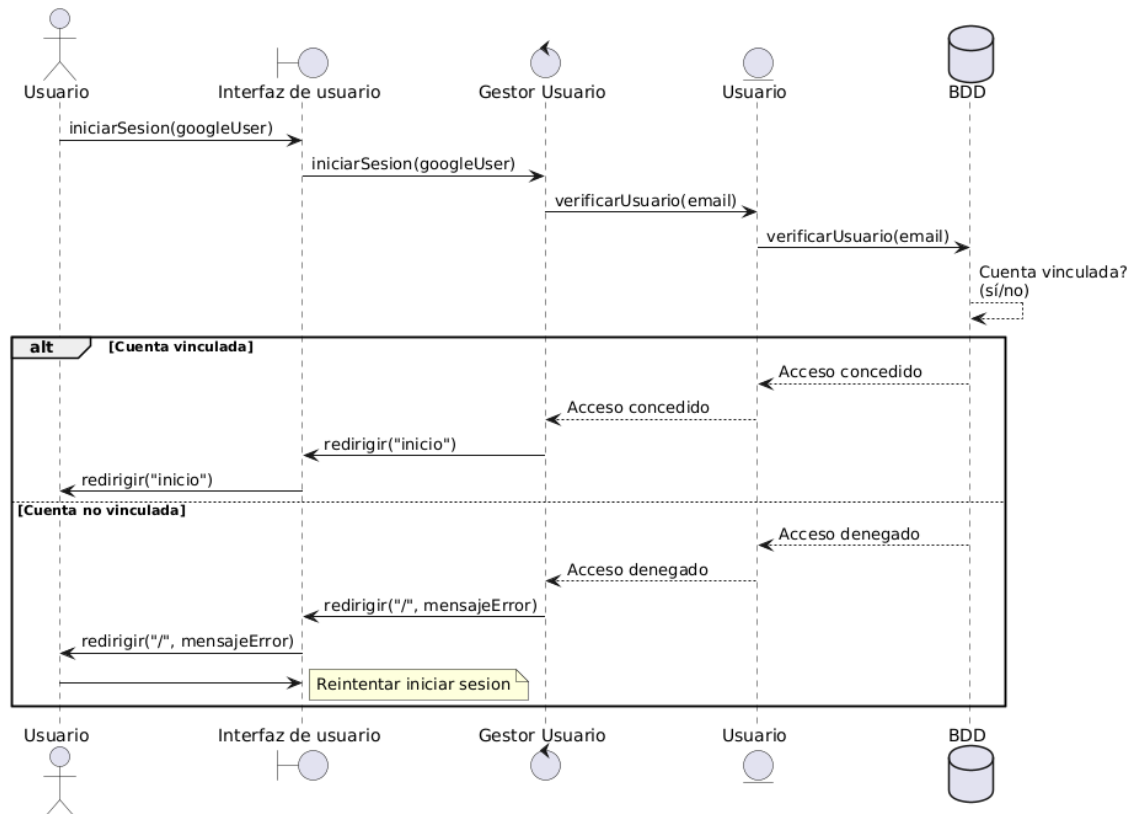
#### Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

- GestorUsuario.
  - Tendrá los métodos: Constructor, retornar un usuario por correo.
  - Comentario: El método obtener usuario tendrá que contar con un método privado que valide que se está pasando un correo.
- Usuario.
  - Tendrá los atributos de nombre, correo y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, retornar un usuario por correo.

Estas clases están vinculada al subsistema de UARGflow.

## Diagrama de Interacción



**Fig. 1.1:** Diagrama de secuencia CU01.

## Requerimientos especiales o de implementación

- **(RNF04):** El sistema deberá ser integrado con UARGflow para el inicio de sesión de los usuarios.

## Caso de Uso 2: Administrar acceso al sistema

### Especificación de caso de uso de diseño

CASO DE USO N°2	Administrar acceso al sistema	
Actor Principal	Administrador	
Precondiciones	El usuario inicio sesión en el sistema	
Disparador	El usuario selecciona una de las opciones de administración de acceso	
DESCRIPCION	Paso	Acción
	1	El actor ingresa a la lista de usuarios.
	2	La <u>InterfazUsuario</u> solicita <i>listarUsuarios</i> al <u>GestorUsuario</u> . El <u>GestorUsuario</u> envía el evento <i>listarUsuarios</i> a <u>Usuario</u> . <u>Usuario</u>

		solicita <i>listarUsuarios</i> a la <u>BDD</u> , la <u>BDD</u> realiza la transacción, la <u>BDD</u> retorna la respuesta al <u>Usuario</u> , el <u>Usuario</u> retorna la respuesta al <u>GestorUsuario</u> .
	3	El <u>GestorUsuario</u> retorna los usuarios a la <u>InterfazUsuario</u> . La <u>InterfazUsuario</u> redirige al actor a la página /usuarios.
	4	El actor selecciona “Registrar usuario” (4a). El actor selecciona “Eliminar usuario” (4b). El actor selecciona “Editar Usuario” (4c).
EXTENSIONES	Paso	Acción de Derivación
	4a	<ol style="list-style-type: none"> <li>1. La <u>InterfazUsuario</u> redirige al actor a la página /usuarios/crear.</li> <li>2. El actor ingresa los datos del usuario a <i>registrar (nombre, email, roles)</i> en el formulario de la página y selecciona confirmar. Si selecciona cancelar, la <u>InterfazUsuario</u> redirige al actor a /usuarios.</li> <li>3. La <u>InterfazUsuario</u> solicita <i>crearUsuario(datos)</i> a <u>GestorUsuario</u>. El <u>GestorUsuario</u> comprueba los datos y envía el evento <i>crearUsuario(datos)</i> a <u>Usuario</u>. <u>Usuario</u> solicita <i>crearUsuario(datos)</i> a la <u>BDD</u>. La <u>BDD</u> carga el usuario y retorna la respuesta al <u>Usuario</u>, el <u>Usuario</u> retorna la respuesta al <u>GestorUsuario</u>, el <u>GestorUsuario</u> retorna la respuesta a la <u>InterfazUsuario</u>. La <u>InterfazUsuario</u> redirige al actor a la página /usuarios.</li> </ol>
	4b	<ol style="list-style-type: none"> <li>1. El actor selecciona el usuario a eliminar.</li> <li>2. La <u>InterfazUsuario</u> redirige al actor a la página /usuarios/eliminar/:id con la id del usuario seleccionado y solicita confirmación.</li> <li>3. El actor confirma la eliminación del usuario. Si cancela la eliminación, la <u>InterfazUsuario</u> redirige al actor a /usuarios.</li> <li>4. La <u>InterfazUsuario</u> envía el evento <i>eliminarUsuario(id_usuario)</i> al <u>GestorUsuario</u>, el <u>GestorUsuario</u> envía el evento <i>eliminarUsuario(id_usuario)</i> a <u>Usuario</u>. <u>Usuario</u> solicita <i>eliminarUsuario(id_usuario)</i> a la <u>BDD</u>. La <u>BDD</u> elimina el usuario. La <u>BDD</u> retorna la respuesta al <u>Usuario</u>, el <u>Usuario</u> retorna la respuesta al <u>GestorUsuario</u>, el <u>GestorUsuario</u> retorna la respuesta a la <u>InterfazUsuario</u>.</li> <li>5. La <u>InterfazUsuario</u> redirige al actor a /usuarios.</li> </ol>



	4c	<ol style="list-style-type: none"> <li>1. El actor selecciona el usuario a modificar.</li> <li>2. La <u>InterfazUsuario</u> redirige al actor a la página /usuarios/editar/:id con la id del usuario seleccionado.</li> <li>3. El actor modifica los datos del usuario seleccionado (nombre, email, roles) en el formulario de la página y selecciona confirmar. Si selecciona cancelar, la <u>InterfazUsuario</u> redirige al actor a /usuarios.</li> <li>4. La <u>InterfazUsuario</u> solicita <i>editarUsuario(id_usuario, datos)</i> a <u>GestorUsuario</u>. El <u>GestorUsuario</u> comprueba los datos y envía el evento <i>editarUsuario(id_usuario, datos)</i> a <u>Usuario</u>. <u>Usuario</u> solicita <i>editarUsuario(id_usuario, datos)</i> a la <u>BDD</u>. La <u>BDD</u> modifica el usuario. La <u>BDD</u> retorna la respuesta al <u>Usuario</u>, el <u>Usuario</u> retorna la respuesta al <u>GestorUsuario</u>, el <u>GestorUsuario</u> retorna a la <u>InterfazUsuario</u>, La <u>InterfazUsuario</u> redirige al actor a /usuarios.</li> </ol>
--	----	---

## Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

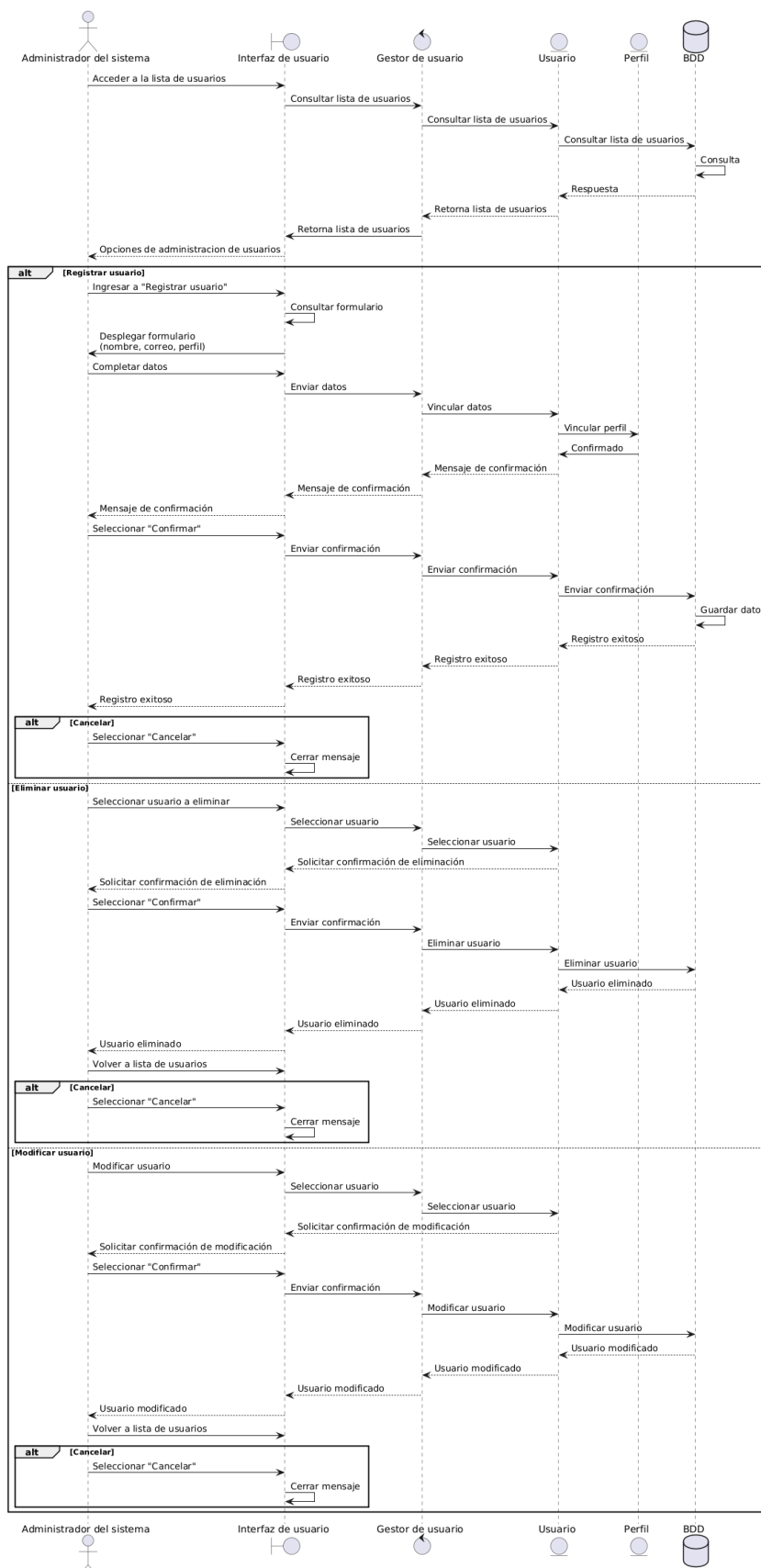
- GestorUsuario.
  - Tendrá los métodos: Constructor, retornar un usuario por correo, retornar todos los usuarios, modificar un usuario, eliminar un usuario, actualizar perfil de usuario, actualizar permiso de perfil, añadir perfil, eliminar perfil.
  - Comentario: El método obtener usuario tendrá que contar con un método privado que valide que se está pasando un correo, Los métodos que se repiten entre usuario y gestión de usuario es para formatear y respetar con la modularización del sistema.
- Usuario.
  - Tendrá los atributos de nombre, correo, conexión y perfil.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, retornar un usuario por correo, retornar todos los usuarios, modificar un usuario, eliminar un usuario, actualizar perfil de usuario.
- Perfil.
  - Tendrá el atributo nombre, conexión y permisos.
  - Tendrá los métodos: Constructor, método get y set de cada atributo excepto para la conexión, obtener todos los perfiles, obtener un perfil por id, eliminar por id el perfil, actualizar por id el perfil, añadir perfil, actualizar permisos en perfil.
- Permiso.
  - Tendrá el atributo nombre y conexión.



- Tendrá el método: Constructor, obtener todos los permisos, obtener permisos de usuario.
- Comentario: Si bien esta clase no aparece en el diagrama de secuencia, esta de manera implícita cuando se habla de administrador, usuario, etc.

Estas clases están vinculadas al subsistema de UARGflow.

## Diagrama de Interacción



**Fig. 1.2:** Diagrama de secuencia CU02.

## Caso de Uso 3: Administrar proyectos

### Especificación de caso de uso de diseño

CASO DE USO N°3	Administrar proyectos	
Actor Principal	Administrador	
Precondiciones	El usuario inicio sesión en el sistema	
Disparador	El usuario selecciona una de las opciones de administración de proyectos	
DESCRIPCION	Paso	Acción
	1	El actor ingresa a la lista de proyectos.
	2	La <u>InterfazUsuario</u> solicita <i>listarProyectos</i> al <u>GestorProyecto</u> . El <u>GestorProyecto</u> envía el evento <i>listarProyectos</i> a <u>Proyecto</u> . <u>Proyecto</u> solicita <i>listarProyectos</i> a la <u>BDD</u> , la <u>BDD</u> realiza la transacción, la <u>BDD</u> retorna la respuesta al <u>Proyecto</u> , el <u>Proyecto</u> retorna la respuesta al <u>GestorProyecto</u> . El <u>GestorProyecto</u> retorna la respuesta a la <u>InterfazUsuario</u> . La <u>InterfazUsuario</u> redirige al actor a la pagina /proyectos.
	3	El actor selecciona "Crear proyecto" (3a). El actor selecciona "Modificar proyecto" (3b).
EXTENSIONES	Paso	Acción de Derivación
	3a	<ol style="list-style-type: none"> <li>1. La <u>InterfazUsuario</u> redirige al actor a la página /proyectos/crear.</li> <li>2. El actor ingresa los datos del proyecto a <i>registrar (nombre, descripcion, estado, participantes, iteraciones, categorias)</i> en el formulario de la página y selecciona confirmar. Si selecciona cancelar, la <u>InterfazUsuario</u> redirige al actor a /proyectos.</li> <li>3. La <u>InterfazUsuario</u> solicita <i>crearProyecto(datos)</i> a <u>GestorProyecto</u>. El <u>GestorProyecto</u> comprueba los datos y envía el evento <i>crearProyecto(datos)</i> a <u>Proyecto</u>. <u>Proyecto</u> solicita <i>crearProyecto(datos)</i> a la <u>BDD</u>. La <u>BDD</u> carga el proyecto y retorna la respuesta al <u>Proyecto</u>, el <u>Proyecto</u> retorna la respuesta al <u>GestorProyecto</u>, el <u>GestorProyecto</u> retorna la respuesta a la <u>InterfazUsuario</u>. La <u>InterfazUsuario</u> redirige al actor a la página /proyectos.</li> </ol>
	3b	<ol style="list-style-type: none"> <li>1. El actor selecciona el proyecto a modificar.</li> </ol>

		<p>2. La <u>InterfazUsuario</u> redirige al actor a la página /proyecto/editar/:id con la id del proyecto seleccionado.</p> <p>3. El actor modifica los datos del usuario seleccionado (nombre, descripción, estado, participantes, iteraciones, categorías) en el formulario de la página y selecciona confirmar. Si selecciona cancelar, la <u>InterfazUsuario</u> redirige al actor a /proyectos.</p> <p>4. La <u>InterfazUsuario</u> solicita <i>editarProyecto(id_proyecto, datos)</i> a <u>GestorProyecto</u>. El <u>GestorProyecto</u> comprueba los datos y envía el evento <i>editarProyecto(id_proyecto, datos)</i> a <u>Proyecto</u>. <u>Proyecto</u> solicita <i>editarProyecto(id_proyecto, datos)</i> a la <u>BDD</u>. La <u>BDD</u> modifica el usuario. La <u>BDD</u> retorna la respuesta al <u>Proyecto</u>, el <u>Proyecto</u> retorna la respuesta al <u>GestorProyecto</u>, el <u>GestorProyecto</u> retorna a la <u>InterfazUsuario</u>, La <u>InterfazUsuario</u> redirige al actor a /usuarios.</p>
--	--	---

## Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

- GestorProyecto.
  - Tendrá los métodos: Constructor, retornar un proyecto por id, retornar proyecto por nombre, crear proyecto, actualizar proyecto, vincular o desvincular iteraciones, obtener todos los proyectos, vincular o desvincular participantes, obtener iteraciones por id de proyecto, actualizar iteración por id, eliminar iteración por id, añadir iteración.
  - Comentario: Cada método deberá comprobar que no se pase ningún campo vacío, ni ninguna inyección sql.
- Proyecto.
  - Tendrá los atributos de nombre, descripción, estado, fecha inicio, fecha fin, iteraciones, participantes y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, retornar un proyecto por id, retornar proyecto por nombre, crear proyecto, actualizar proyecto por id, actualizar iteraciones, obtener todos los proyectos, vincular o desvincular participantes.
- Iteracion.
  - Tendrá los atributos fecha inicio, fecha fin y conexión.
  - Tendrá los métodos: Constructor, método get y set de cada atributo excepto para la conexión, obtener iteraciones por id de proyecto, actualizar iteración por id, eliminar iteración por id, añadir iteración.

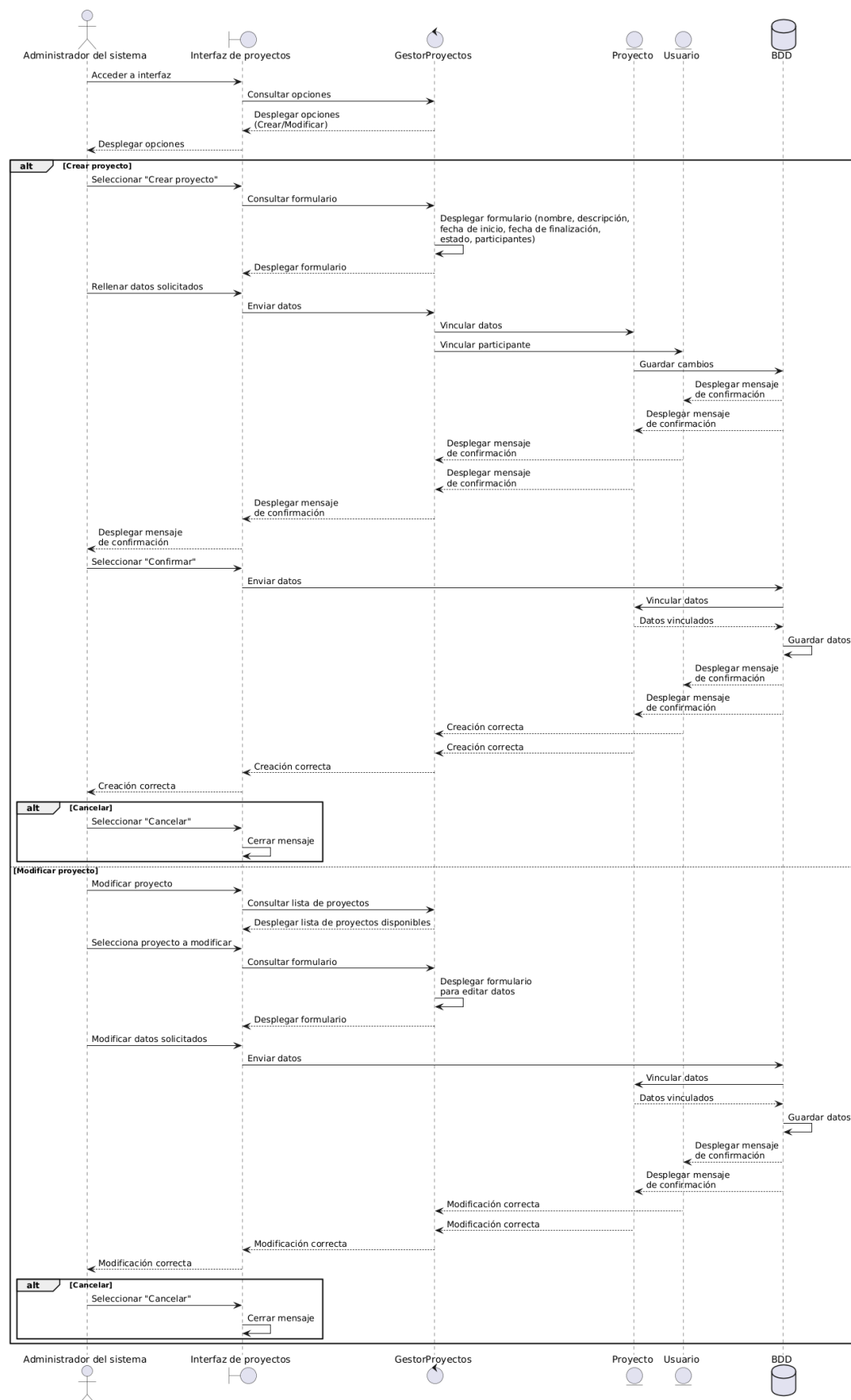


Estas clases están vinculadas al subsistema de Proyectos.

- GestorUsuario.
  - Tendrá el método: Constructor, obtener todos los usuarios.
- Usuario
  - Tendrá los atributos nombre, correo y conexión.
  - Tendrá los métodos: Constructor, obtener todos los usuarios.

Estas clases están vinculadas al subsistema de UARGflow.

## Diagrama de Interacción



**Fig. 1.3:** Diagrama de secuencia CU03.

## Caso de Uso 4: Añadir riesgo a la lista

### Especificación de caso de uso de diseño

CASO DE USO N°4			Añadir riesgo a la lista
Actor Principal	Líder de proyecto y Desarrollador		
Precondiciones	El usuario inicio sesión en el sistema. El usuario está vinculado a un proyecto		
Disparador	El usuario ingresa al proyecto y selecciona añadir riesgo.		
DESCRIPCION	Paso	Acción	
	1	El actor selecciona "Añadir riesgo".	
	2	La <u>InterfazUsuario</u> redirige al actor a la página /proyecto/:nombre/riesgo/crear/.	
	3	El actor ingresa los datos del riesgo (descripción, responsable, categoría) en el formulario de la página. El actor selecciona confirmar. Si selecciona cancelar, la <u>InterfazUsuario</u> redirige al actor a la página /proyecto/:nombre/riesgo.	
	4	La <u>InterfazUsuario</u> solicita <i>crearRiesgo(datos)</i> a <u>GestorRiesgo</u> . El <u>GestorRiesgo</u> envia el evento <i>crearRiesgo(datos)</i> a <u>Riesgo</u> . <u>Riesgo</u> solicita <i>crearRiesgo(datos)</i> a la <u>BDD</u> . La <u>BDD</u> carga el riesgo. La <u>BDD</u> retorna la respuesta a <u>Riesgo</u> . El <u>Riesgo</u> retorna la respuesta al <u>GestorRiesgo</u> , el <u>GestorRiesgo</u> retorna la respuesta a la <u>InterfazUsuario</u> . La <u>InterfazUsuario</u> redirige al actor a /proyecto/:nombre/riesgo.	
EXTENSIONES	Paso	Acción de Derivación	

### Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

- GestorRiesgo.
  - Tendrá los métodos: Constructor, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, vincular o desvincular responsable, actualizar categoría de un riesgo, vincular o desvincular iteraciones, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.
- Riesgo.
  - Tendrá los atributos: descripción, factor riesgo, categoría, iteración y conexión.



- Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, actualizar categoría de un riesgo, vincular o desvincular proyecto, vincular o desvincular iteraciones.
- Categoría
  - Tendrá los atributos: nombre, descripción y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.
  - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia.

Estas clases están vinculadas al subsistema de Riesgos.

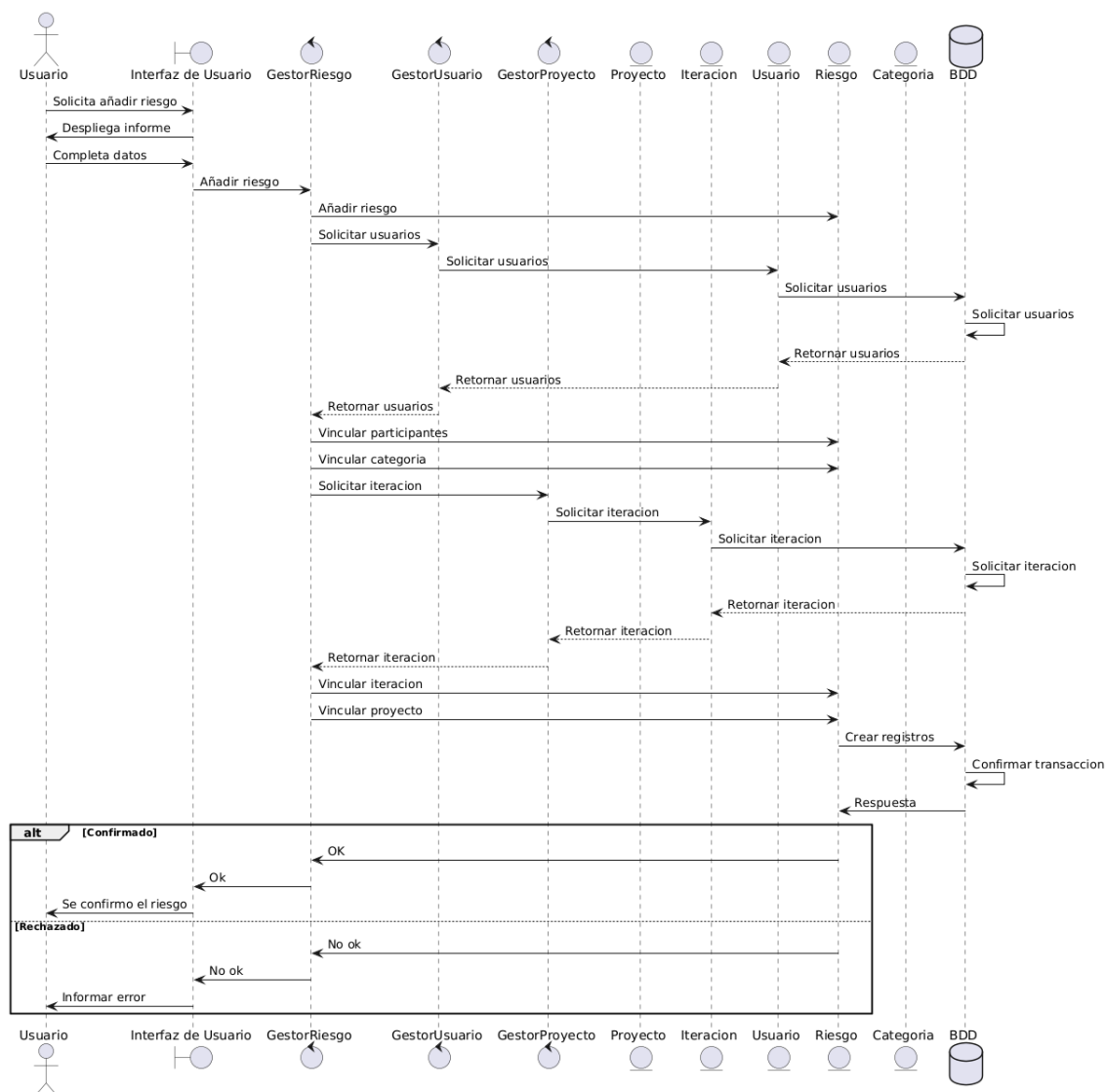
- GestorProyecto.
  - Tendrá los métodos: Constructor, retornar un proyecto por id, retornar proyecto por nombre, crear proyecto, actualizar proyecto, vincular o desvincular iteraciones, obtener todos los proyectos, vincular o desvincular participantes, obtener iteraciones por id de proyecto, actualizar iteración por id, eliminar iteración por id, añadir iteración.
  - Comentario: Cada método deberá comprobar que no se pase ningún campo vacío, ni ninguna inyección sql.
- Proyecto.
  - Tendrá los atributos de nombre, descripción, estado, fecha inicio, fecha fin, iteraciones, participantes y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, retornar un proyecto por id, retornar proyecto por nombre, crear proyecto, actualizar proyecto, actualizar iteraciones, obtener todos los proyectos, vincular o desvincular participantes.
- Iteracion.
  - Tendrá los atributos fecha inicio, fecha fin y conexión.
  - Tendrá los métodos: Constructor, método get y set de cada atributo excepto para la conexión, obtener iteraciones por id de proyecto, actualizar iteración por id, eliminar iteración por id, añadir iteración.

Estas clases están vinculadas al subsistema de Proyectos.

- GestorUsuario.
  - Tendrá el método: Constructor, obtener todos los usuarios.
- Usuario
  - Tendrá los atributos nombre, correo y conexión.
  - Tendrá los métodos: Constructor, obtener todos los usuarios.

Estas clases están vinculadas al subsistema de UARGflow.

## Diagrama de Interacción



**Fig. 1.4:** Diagrama de secuencia CU04.

## Caso de Uso 5: Modificar lista de riesgos

### Especificación de caso de uso de diseño

CASO DE USO N°5		Modificar lista de riesgos
Actor Principal	Líder del proyecto	
Precondiciones	1. El usuario realizó el Caso de uso 1. 2. El usuario está vinculado a un proyecto. 3. Existe al menos un riesgo cargado.	
Disparador	El usuario selecciona la opción "Modificar lista de riesgos".	
DESCRIPCION	Paso	Acción
	1	El actor selecciona la opción "Modificar lista de riesgos".

	2	La <u>InterfazUsuario</u> solicita <i>mostrarFormularioEdicion</i> al <u>GestorRiesgos</u> .
	3	El <u>GestorRiesgos</u> envía el evento <i>obtenerDatosRiesgo</i> a <u>Riesgo</u> .
	4	<u>Riesgo</u> solicita <i>obtenerDatosRiesgo</i> a la <u>BDD</u> . La <u>BDD</u> realiza la transacción y retorna los datos al <u>Riesgo</u> , el <u>Riesgo</u> retorna la respuesta al <u>GestorRiesgos</u> .
	5	El <u>GestorRiesgos</u> retorna el formulario a la <u>InterfazUsuario</u> .
	6	El actor modifica los datos permitidos (responsable y categoría) y selecciona "Guardar" (6a). Si selecciona "Cancelar", continúa con la extensión (6b).
	7	La <u>InterfazUsuario</u> solicita <i>actualizarRiesgo(id_riesgo, datos)</i> al <u>GestorRiesgos</u> . El <u>GestorRiesgos</u> valida los datos y envía el evento <i>actualizarRiesgo(id_riesgo, datos)</i> a <u>Riesgo</u> . <u>Riesgo</u> solicita <i>actualizarRiesgo(id_riesgo, datos)</i> a la <u>BDD</u> . La <u>BDD</u> actualiza el riesgo y retorna la respuesta al <u>Riesgo</u> , el <u>Riesgo</u> retorna la respuesta al <u>GestorRiesgos</u> .
	8	El <u>GestorRiesgos</u> envía la confirmación a la <u>InterfazUsuario</u> y esta muestra un mensaje de éxito al actor.
EXTENSIONES	Paso	Acción de Derivación
	6a	<ol style="list-style-type: none"> <li>1. El <u>GestorRiesgos</u> detecta campos obligatorios sin completar.</li> <li>2. El <u>GestorRiesgos</u> envía los errores a la <u>InterfazUsuario</u>.</li> <li>3. La <u>InterfazUsuario</u> muestra los mensajes de error al actor.</li> <li>4. El flujo continúa en el paso 3 de la descripción principal.</li> </ol>
	6b	<ol style="list-style-type: none"> <li>1. La <u>InterfazUsuario</u> cierra el formulario de edición.</li> <li>2. La <u>InterfazUsuario</u> retorna a la vista previa de la lista de riesgos.</li> </ol>

## Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

- GestorRiesgo.
  - Tendrá los métodos: Constructor, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, vincular o desvincular responsable, actualizar categoría de un riesgo, vincular o desvincular iteraciones, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id.
- Riesgo.
  - Tendrá los atributos: descripción, factor riesgo, categoría, iteración y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id,

actualizar categoría de un riesgo, vincular o desvincular proyecto, vincular o desvincular iteraciones.

- Categoría
  - Tendrá los atributos: nombre, descripción y conexión
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.
  - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia.
- Evaluacion
  - Tendrá los atributos: impacto, probabilidad y conexión
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id.

Estas clases están vinculadas al subsistema de Riesgos.

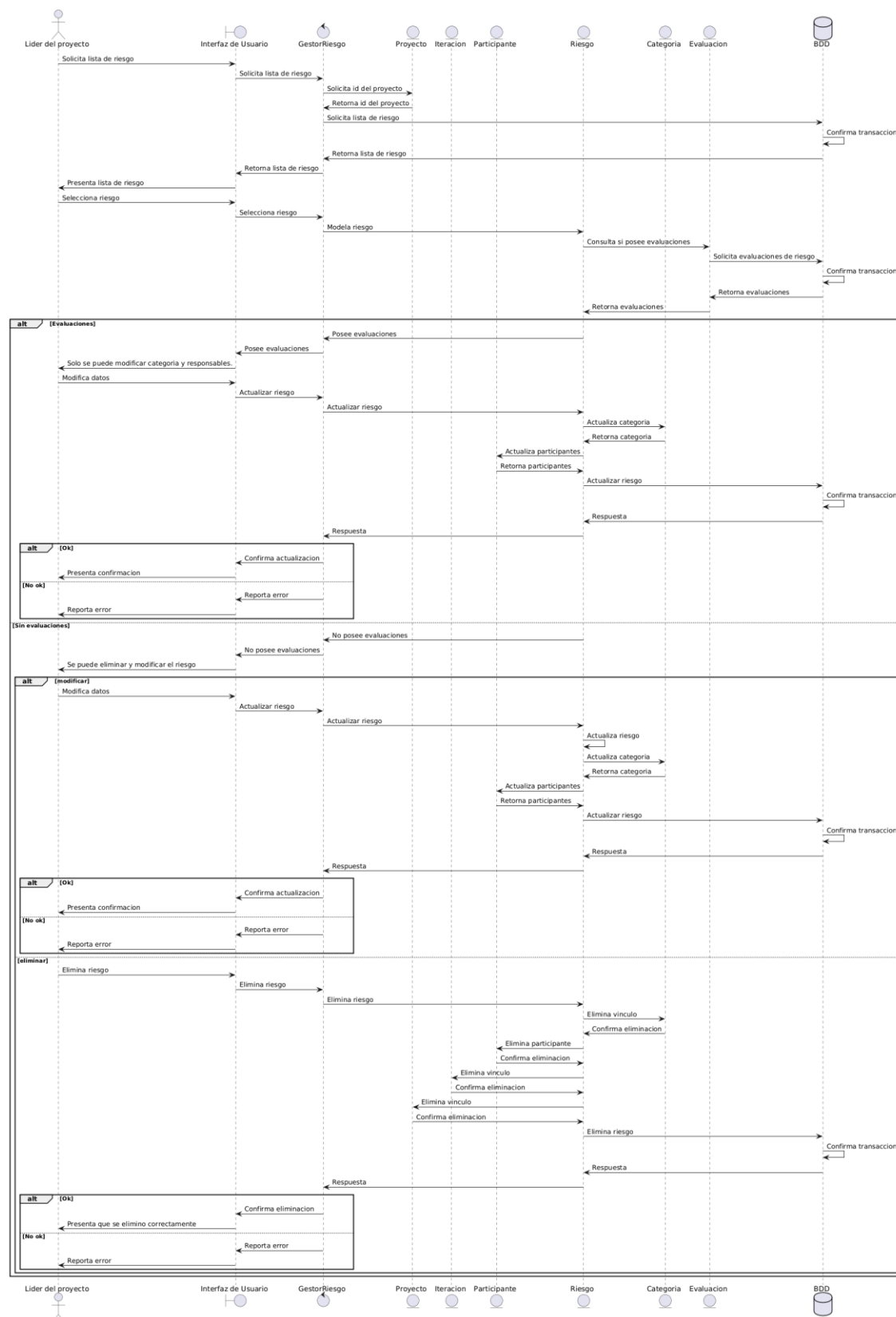
- GestorProyecto.
  - Tendrá los métodos: Constructor, retornar un proyecto por id, retornar proyecto por nombre, crear proyecto, actualizar proyecto, vincular o desvincular iteraciones, obtener todos los proyectos, vincular o desvincular participantes, obtener iteraciones por id de proyecto, actualizar iteración por id, eliminar iteración por id, añadir iteración.
  - Comentario: Cada método deberá comprobar que no se pase ningún campo vacío, ni ninguna inyección sql.
- Proyecto.
  - Tendrá los atributos de nombre, descripción, estado, fecha inicio, fecha fin, iteraciones, participantes y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, retornar un proyecto por id, retornar proyecto por nombre, crear proyecto, actualizar proyecto, actualizar iteraciones, obtener todos los proyectos, vincular o desvincular participantes.
- Iteracion.
  - Tendrá los atributos fecha inicio, fecha fin y conexión.
  - Tendrá los métodos: Constructor, método get y set de cada atributo excepto para la conexión, obtener iteraciones por id de proyecto, actualizar iteración por id, eliminar iteración por id, añadir iteración.

Estas clases están vinculadas al subsistema de Proyectos.

- GestorUsuario.
  - Tendrá el método: Constructor, obtener todos los usuarios.
- Usuario
  - Tendrá los atributos nombre, correo y conexión.
  - Tendrá los métodos: Constructor, obtener todos los usuarios.

Estas clases están vinculadas al subsistema de UARGflow.

## Diagrama de Interacción



**Fig. 1.5:** Diagrama de secuencia CU05.

## Caso de Uso 6: Administrar categorías de riesgos

### Especificación de caso de uso de diseño

CASO DE USO N°6 <Nombre CU>		
Actor Principal	<Actores>	
Precondiciones	<Condiciones que se deben cumplir antes de llamar al CU>	
Disparador	<Accion que inicia el CU>	
DESCRIPCION	Paso	Acción
EXTENSIONES	Paso	Acción de Derivación

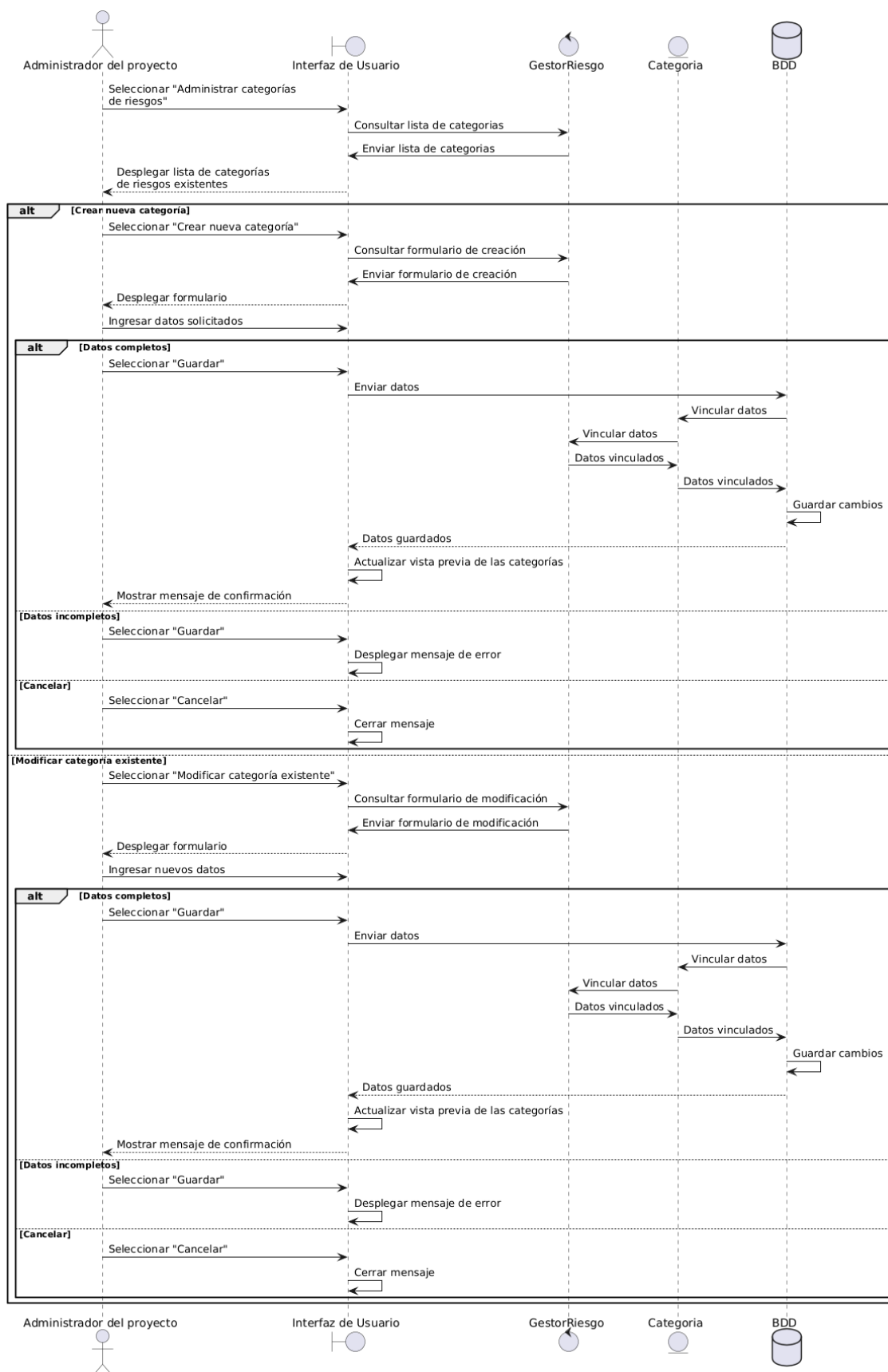
### Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

- GestorRiesgo.
  - o Tendrá los métodos: Constructor, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, vincular o desvincular responsable, actualizar categoría de un riesgo, vincular o desvincular iteraciones, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.
- Riesgo.
  - o Tendrá los atributos: descripción, factor riesgo, categoría, iteración y conexión.
  - o Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, actualizar categoría de un riesgo, vincular o desvincular proyecto, vincular o desvincular iteraciones.
- Categoria
  - o Tendrá los atributos: nombre, descripción y conexión.
  - o Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.

Estas clases están vinculadas al subsistema de Riesgos.

## Diagrama de Interacción



**Fig. 1.6** Diagrama de secuencia CU06.

## Caso de Uso 7: Realizar evaluación de riesgo

### Especificación de caso de uso de diseño

CASO DE USO N°7 Realizar evaluación de riesgo		
Actor Principal	Líder del proyecto; Desarrollador	
Precondiciones	<ul style="list-style-type: none"> <li>El usuario completó el Caso de uso 1.</li> <li>El usuario está vinculado a un proyecto.</li> <li>Existe al menos un riesgo cargado.</li> </ul>	
Disparador	El usuario selecciona la opción "Realizar evaluación" en la lista de riesgos.	
DESCRIPCION	Paso	Acción
	1	El actor selecciona la opción "Realizar evaluación" en la lista de riesgos.
	2	La <u>InterfazRiesgos</u> solicita <i>listarRiesgos</i> al <u>GestorRiesgos</u> .
	3	El <u>GestorRiesgos</u> envía el evento <i>listarRiesgos</i> a <u>Riesgo</u> .
	4	<u>Riesgo</u> solicita <i>listarRiesgos</i> a la <u>BDD</u> , la <u>BDD</u> realiza la transacción y retorna la respuesta a <u>Riesgo</u> , <u>Riesgo</u> retorna la respuesta al <u>GestorRiesgos</u> .
	5	El <u>GestorRiesgos</u> retorna los riesgos a la <u>InterfazRiesgos</u> . La <u>InterfazRiesgos</u> muestra la lista de riesgos al actor.
	6	El actor selecciona el riesgo que desea evaluar.
	7	El actor selecciona el riesgo que desea evaluar y selecciona "Evaluar" (7a) o "Cancelar" (7b).
EXTENSIONES	Paso	Acción de Derivación
	7a	<ol style="list-style-type: none"> <li>La <u>InterfazRiesgos</u> solicita <i>actualizarRiesgo(datos)</i> al <u>GestorRiesgos</u>.</li> <li>El <u>GestorRiesgos</u> valida los datos (7a1) y envía el evento <i>actualizarRiesgo(datos)</i> a <u>Riesgo</u>.</li> <li><u>Riesgo</u> solicita <i>actualizarRiesgo(datos)</i> a la <u>BDD</u>. La <u>BDD</u> actualiza el factor de riesgo.</li> <li>La <u>BDD</u> retorna la respuesta a <u>Riesgo</u>, <u>Riesgo</u> retorna la respuesta al <u>GestorRiesgos</u>.</li> <li>El <u>GestorRiesgos</u> retorna la respuesta a la <u>InterfazRiesgos</u>.</li> <li>La <u>InterfazRiesgos</u> redirige al Caso de uso 12.</li> <li>La <u>InterfazRiesgos</u> muestra un mensaje de confirmación de que la evaluación ha sido añadida exitosamente.</li> </ol>
	7a1	<ol style="list-style-type: none"> <li>Si el usuario no rellenó correctamente los datos obligatorios del campo, el <u>GestorRiesgos</u> retorna los errores.</li> <li>La <u>InterfazRiesgos</u> muestra los mensajes indicando los errores cometidos por el usuario.</li> <li>Vuelve al paso 7.</li> </ol>
	7b	<ol style="list-style-type: none"> <li>La <u>InterfazRiesgos</u> cancela la operación y cierra el mensaje.</li> </ol>

### Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

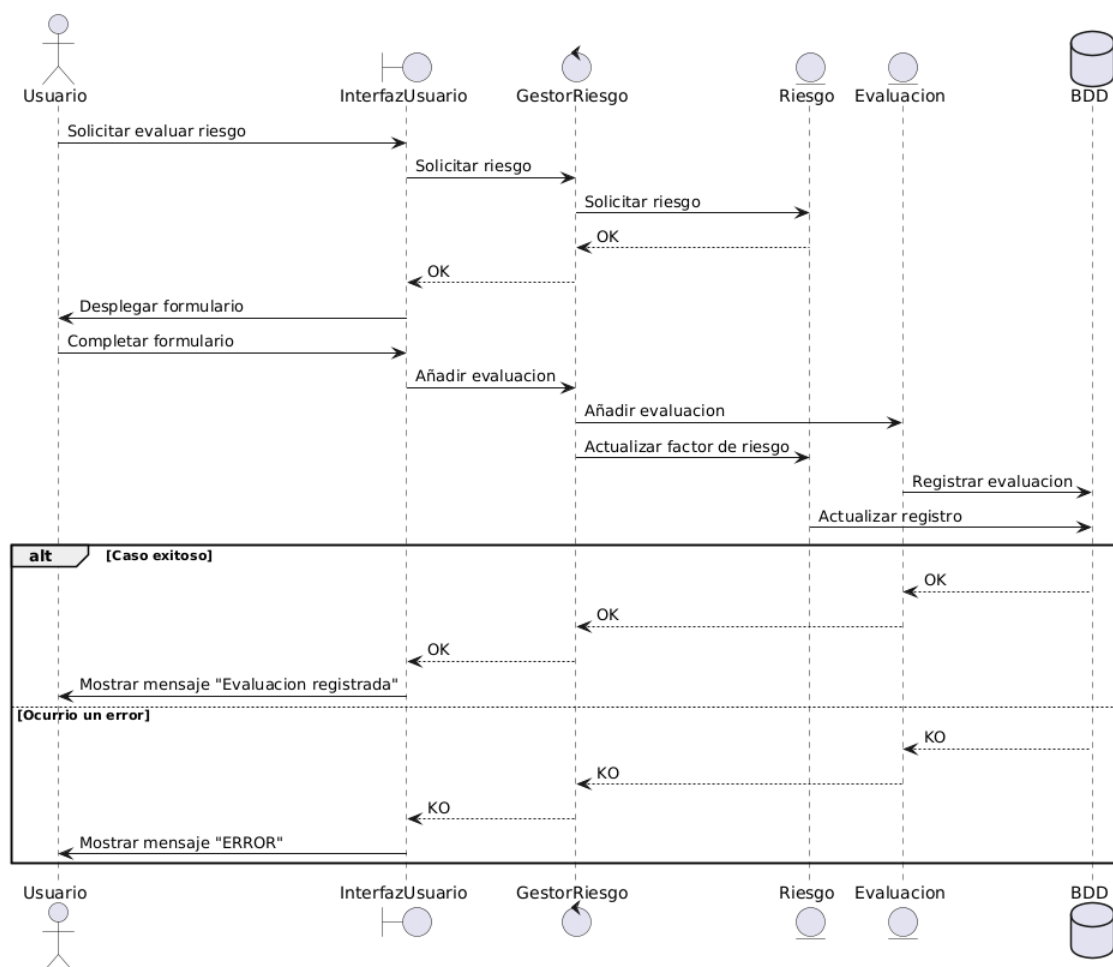
- GestorRiesgo.



- Tendrá los métodos: Constructor, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, vincular o desvincular responsable, actualizar categoría de un riesgo, vincular o desvincular iteraciones, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id.
- Riesgo.
  - Tendrá los atributos: descripción, factor riesgo, categoría, iteración y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, actualizar categoría de un riesgo, vincular o desvincular proyecto, vincular o desvincular iteraciones.
- Categoría
  - Tendrá los atributos: nombre, descripción y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.
  - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia.
- Evaluacion
  - Tendrá los atributos: impacto, probabilidad y conexión
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id.

Estas clases están vinculadas al subsistema de Riesgos.

## Diagrama de Interacción



**Fig. 1.7:** Diagrama de secuencia CU07.

## Caso de Uso 8: Añadir plan de riesgo

### Especificación de caso de uso de diseño

CASO DE USO N°8	Añadir plan de riesgo	
Actor Principal	Líder del proyecto; Desarrollador	
Precondiciones	<ul style="list-style-type: none"> <li>El actor completó el Caso de uso 1.</li> <li>El actor está vinculado a un proyecto.</li> <li>Existe al menos un riesgo cargado.</li> </ul>	
Disparador	El usuario selecciona la opción "Realizar plan de riesgo" en la lista de riesgos.	
DESCRIPCION	Paso	Acción
	1	El actor solicita añadirPlan(idRiesgo) a la InterfazUsuario. La InterfazUsuario redirige al usuario a la página /proyecto/:nombre/riesgos/agregarPlan.
	2	El actor rellena el formulario de la página con los datos del plan de riesgo (id, tipo, descripción, tareas) (Excepción 2a) (Extensión 2b).
	3	La InterfazUsuario solitica añadirPlan(idRiesgo, datosPlan) a GestorRiesgo. GestorRiesgo envia el evento añadirPlan(datosPlan) a Riesgo. Riesgo solicita actualizarRiesgo() a la BDD. La InterfazUsuario muestra un mensaje al usuario confirmando que se añadió el plan de

		riesgo. El actor selecciona confirmar. La InterfazUsuario redirige al actor a /proyecto/:nombre/riesgos
EXTENSIONES	Paso	Acción de Derivación
	2a	El actor completo el formulario con campos inválidos (ninguna tarea agregada (debe haber al menos una), campos vacíos o tipo de plan invalido). La InterfazUsuario muestra un mensaje de error e indica los campos que deben ser corregidos.
	2b	El actor selecciona la opción "Agregar tarea" desde el formulario. La InterfazUsuario despliega un formulario para ingresar los datos de la tarea. El actor completa los datos. La InterfazUsuario solicita agregarTarea(datosTarea) al GestorRiesgo. El GestorRiesgo envia el evento añadirTareas() a PlanRiesgo. La InterfazUsuario devuelve al actor al formulario de plan de riesgos (Excepción 2bi).
	2bi	El actor completo el formulario de tarea con campos invalidos (nombre de longitud mayor a 30 caracteres, responsable inexistente o no participante del proyecto, campos vacíos, fechas de inicio-fin superpuestas, fechas de inicio-fin fuera de la iteración actual). La InterfazUsuario muestra un mensaje de error e indica los campos que deben ser corregidos.

## Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

- GestorRiesgo.
  - o Tendrá los métodos: Constructor, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, vincular o desvincular responsable, actualizar categoría de un riesgo, vincular o desvincular iteraciones, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id, Crear tarea, modificar tarea por id, vincular o desvincular participante, eliminar tarea por id, Crear plan de riesgo, actualizar plan de riesgo por id, eliminar plan de riesgo por id, vincular iteración, Vincular o desvincular tareas.
- Riesgo.
  - o Tendrá los atributos: descripción, factor riesgo, categoría, iteración y conexión.
  - o Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, actualizar categoría de un riesgo, vincular o desvincular proyecto, vincular o desvincular iteraciones.
- Categoría
  - o Tendrá los atributos: nombre, descripción y conexión.
  - o Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.

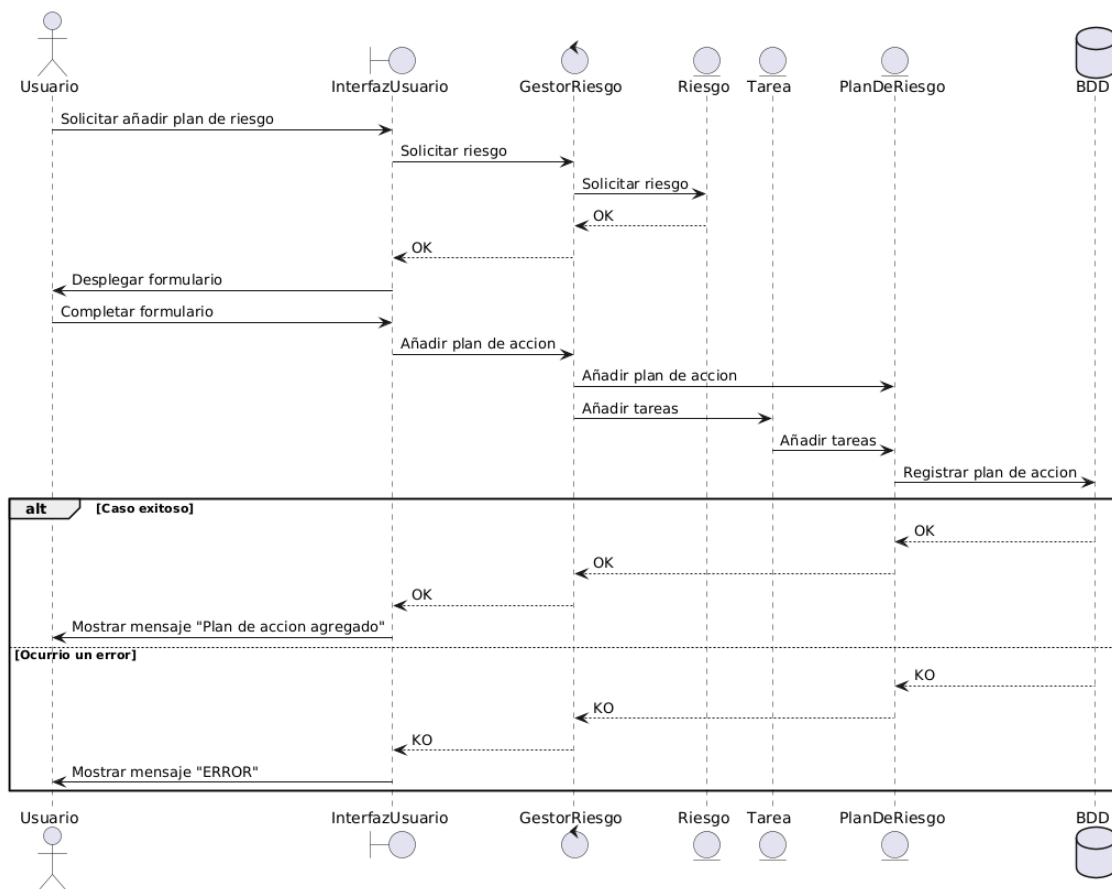
- Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia.
- Evaluacion
  - Tendrá los atributos: impacto, probabilidad y conexión
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id.
  - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia
- PlanDeRiesgo
  - Tendrá los atributos: nombre, descripción, tipo, tareas y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Crear plan de riesgo, actualizar plan de riesgo por id, eliminar plan de riesgo por id, vincular iteración, Vincular o desvincular tareas.
- Tareas:
  - Tendrá los atributos: nombre, descripción, estado, fecha inicio, fecha fin y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Crear tarea, modificar tarea por id, vincular o desvincular participante, eliminar tarea por id.

Estas clases están vinculadas al subsistema de Riesgos.

- GestorUsuario.
  - Tendrá el método: Constructor, obtener todos los usuarios.
- Usuario
  - Tendrá los atributos nombre, correo y conexión.
  - Tendrá los métodos: Constructor, obtener todos los usuarios.

Estas clases están vinculadas al subsistema de UARGflow.

## Diagrama de Interacción



**Fig. 1.8:** Diagrama de secuencia CU08.

## Caso de Uso 9: Modificar plan de acción

### Especificación de caso de uso de diseño

CASO DE USO N°9 Modificar plan de acción		
Actor Principal	Líder del proyecto	
Precondiciones	1. El usuario realizó el Caso de uso 1. 2. El usuario está vinculado a un proyecto. 3. Existe al menos un riesgo cargado.	
Disparador	El usuario selecciona la opción "Modificar plan de acción" en la lista de riesgos.	
DESCRIPCION	Paso	Acción
	1	El actor selecciona la opción "Modificar plan de acción" en la lista de riesgos.
	2	La <u>InterfazUsuario</u> solicita <i>mostrarFormularioPlanAccion</i> al <u>GestorPlanes</u> .
	3	El <u>GestorPlanes</u> envía el evento <i>obtenerDatosPlan</i> a <u>PlanAccion</u> . <u>PlanAccion</u> solicita <i>obtenerDatosPlan</i> a la <u>BDD</u> . La <u>BDD</u> realiza la transacción y retorna los datos al <u>PlanAccion</u> , el <u>PlanAccion</u> retorna la respuesta al <u>GestorPlanes</u> . El <u>GestorPlanes</u> retorna el formulario a la <u>InterfazUsuario</u> .

	4	El actor modifica los datos del plan y selecciona "Guardar" (4a). Si selecciona "Cancelar", continúa con la extensión (4b).
	5	La <u>InterfazUsuario</u> solicita <i>actualizarPlan(id_plan, datos)</i> al <u>GestorPlanes</u> . El <u>GestorPlanes</u> valida los datos y envía el evento <i>actualizarPlan(id_plan, datos)</i> a <u>PlanAccion</u> . <u>PlanAccion</u> solicita <i>actualizarPlan(id_plan, datos)</i> a la <u>BDD</u> . La <u>BDD</u> actualiza el plan y retorna la respuesta al <u>PlanAccion</u> , el <u>PlanAccion</u> retorna la respuesta al <u>GestorPlanes</u> .
	6	El <u>GestorPlanes</u> envía la confirmación a la <u>InterfazUsuario</u> y esta muestra un mensaje de éxito al actor.
EXTENSIONES	Paso	Acción de Derivación
	4a	<ol style="list-style-type: none"> <li>1. El <u>GestorPlanes</u> detecta campos obligatorios sin completar.</li> <li>2. El <u>GestorPlanes</u> envía los errores a la <u>InterfazUsuario</u>.</li> <li>3. La <u>InterfazUsuario</u> muestra los mensajes de error al actor.</li> <li>4. El flujo continúa en el paso 4 de la descripción principal.</li> </ol>
	4b	<ol style="list-style-type: none"> <li>1. La <u>InterfazUsuario</u> cierra el formulario de edición.</li> <li>2. La <u>InterfazUsuario</u> retorna a la vista previa de la lista de riesgos.</li> </ol>

## Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

- GestorRiesgo.
  - o Tendrá los métodos: Constructor, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, vincular o desvincular responsable, actualizar categoría de un riesgo, vincular o desvincular iteraciones, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id, Crear tarea, modificar tarea por id, vincular o desvincular participante, eliminar tarea por id, Crear plan de riesgo, actualizar plan de riesgo por id, eliminar plan de riesgo por id, vincular iteración, Vincular o desvincular tareas.
- Riesgo.
  - o Tendrá los atributos: descripción, factor riesgo, categoría, iteración y conexión.
  - o Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, actualizar categoría de un riesgo, vincular o desvincular proyecto, vincular o desvincular iteraciones.
- Categoria
  - o Tendrá los atributos: nombre, descripción y conexión.

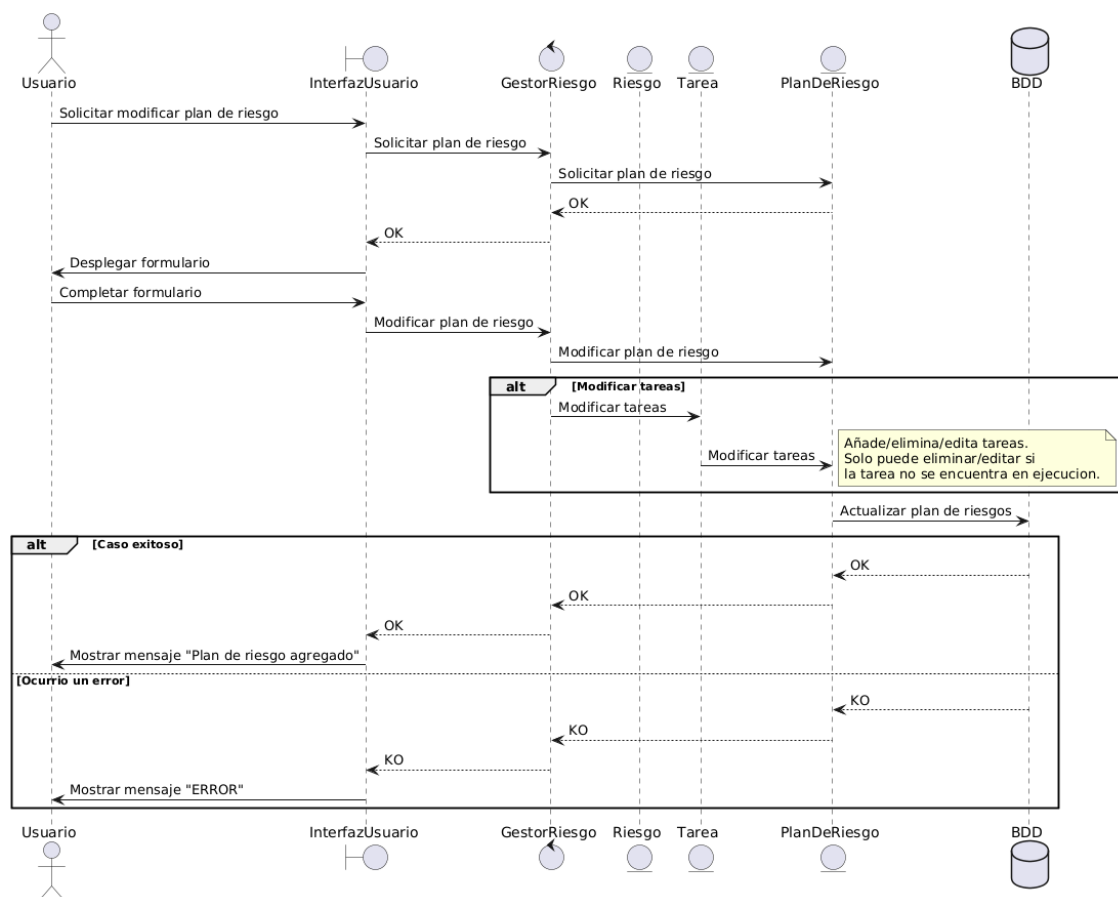
- Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.
- Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia.
- Evaluacion
  - Tendrá los atributos: impacto, probabilidad y conexión
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id.
  - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia
- PlanDeRiesgo
  - Tendrá los atributos: nombre, descripción, tipo, tareas y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Crear plan de riesgo, actualizar plan de riesgo por id, eliminar plan de riesgo por id, vincular iteración, Vincular o desvincular tareas.
- Tareas:
  - Tendrá los atributos: nombre, descripción, estado, fecha inicio, fecha fin y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Crear tarea, modificar tarea por id, vincular o desvincular participante, eliminar tarea por id.

Estas clases están vinculadas al subsistema de Riesgos.

- GestorUsuario.
  - Tendrá el método: Constructor, obtener todos los usuarios.
- Usuario
  - Tendrá los atributos nombre, correo y conexión.
  - Tendrá los métodos: Constructor, obtener todos los usuarios.

Estas clases están vinculadas al subsistema de UARGflow.

## Diagrama de Interacción



**Fig. 1.9:** Diagrama de secuencia CU09.

## Caso de Uso 10: Realizar informes

### Especificación de caso de uso de diseño

CASO DE USO N°	<Nombre CU>	
Actor Principal	<Actores>	
Precondiciones	<Condiciones que se deben cumplir antes de llamar al CU>	
Disparador	<Acción que inicia el CU>	
DESCRIPCION	Paso	Acción
EXTENSIONES	Paso	Acción de Derivación



## Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

- GestorRiesgo.
  - Tendrá los métodos: Constructor, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, vincular o desvincular responsable, actualizar categoría de un riesgo, vincular o desvincular iteraciones, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id, Crear tarea, modificar tarea por id, vincular o desvincular participante, eliminar tarea por id, Crear plan de riesgo, actualizar plan de riesgo por id, eliminar plan de riesgo por id, vincular iteración, Vincular o desvincular tareas, crear incidencia, obtener incidencias por id, obtener todas las incidencias.
- Riesgo.
  - Tendrá los atributos: descripción, factor riesgo, categoría, iteración y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, actualizar categoría de un riesgo, vincular o desvincular proyecto, vincular o desvincular iteraciones.
- Categoria
  - Tendrá los atributos: nombre, descripción y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.
  - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia.
- Evaluacion
  - Tendrá los atributos: impacto, probabilidad y conexión
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id.
  - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia
- PlanDeRiesgo
  - Tendrá los atributos: nombre, descripción, tipo, tareas y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Crear plan de riesgo, actualizar plan de riesgo por id, eliminar plan de riesgo por id, vincular iteración, Vincular o desvincular tareas.
- Tareas:

- Tendrá los atributos: nombre, descripción, estado, fecha inicio, fecha fin y conexión.
- Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Crear tarea, modificar tarea por id, vincular o desvincular participante, eliminar tarea por id.
- Incidencia:
  - Tendrá los atributos: descripción y conexión.
  - Tendrá los métodos: Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, crear incidencia, obtener incidencias por id de riesgo, obtener todas las incidencias.
  - Comentario: Es un tipo de informe.

Estas clases están vinculadas al subsistema de Riesgos.

- GestorArchivos.
  - Tendrá los métodos: generar un informe, obtener informe de incidencia por id, descargar archivo como pdf, descargar archivo como xlsx.
- Informe
  - Tendrá los atributos: datos.
  - Tendrá los métodos: Constructor, los métodos get y set de cada uno de los atributos, generar un informe, obtener informe de incidencia por id.
- Archivo
  - Tendrá el atributo: informe.
  - Tendrá los métodos: Constructor, método get y set del atributo informe, descargar archivo como pdf, descargar archivo como xlsx.

Estas clases están vinculadas al subsistema de Exportar.

### Diagrama de Interacción



**Fig. 1.10:** Diagrama de secuencia CU10.

## Caso de Uso 11: Exportar archivos

### Especificación de caso de uso de diseño

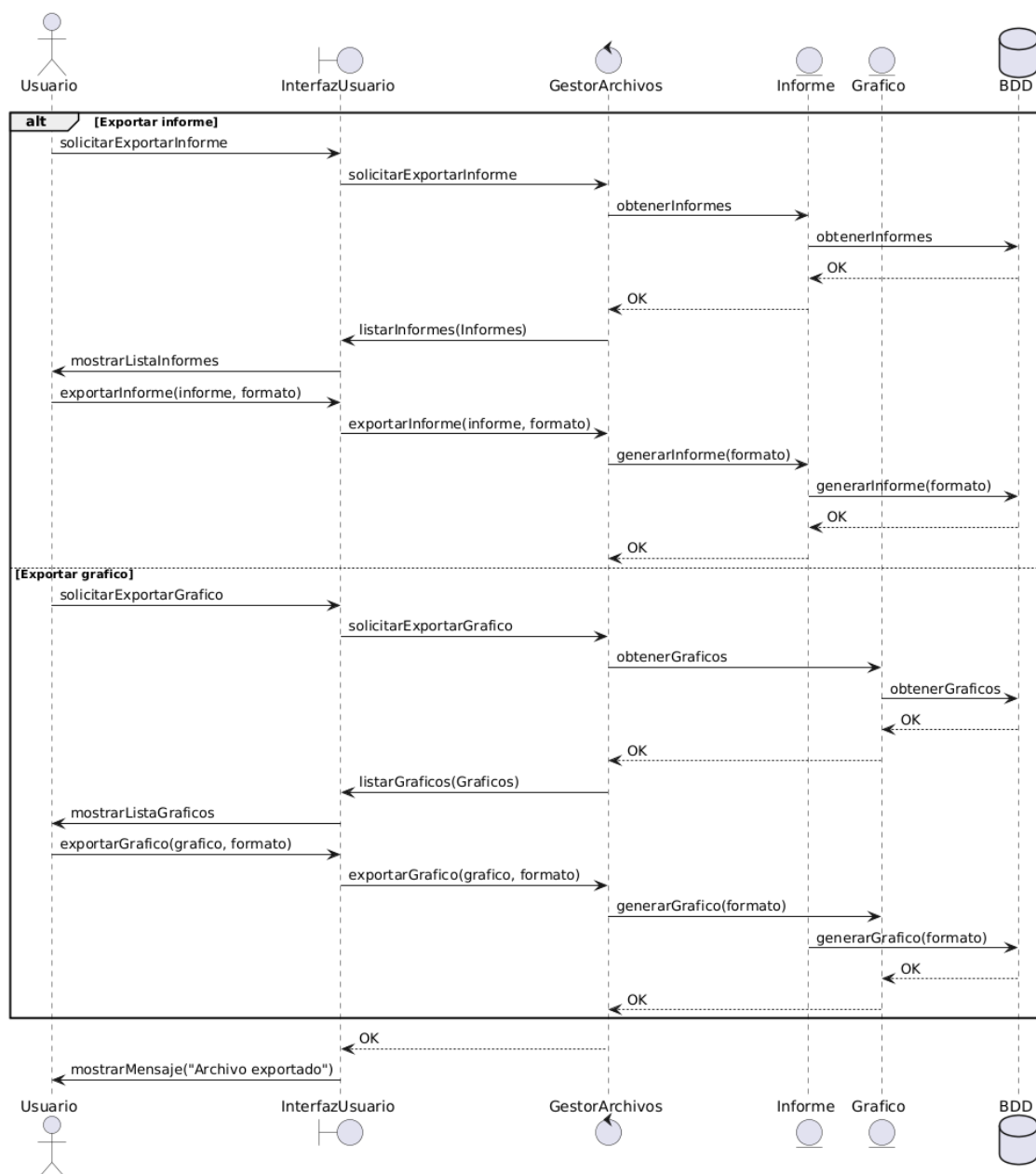
CASO DE USO N°	<Nombre CU>	
Actor Principal	<Actores>	
Precondiciones	<Condiciones que se deben cumplir antes de llamar al CU>	
Disparador	<Accion que inicia el CU>	
DESCRIPCION	Paso	Acción
EXTENSIONES	Paso	Acción de Derivación

### Diagrama de paquetes

- GestorArchivos.
  - Tendrá los métodos: generar un informe, obtener informe de incidencia por id, descargar archivo como pdf, descargar archivo como xlsx.
- Informe
  - Tendrá los atributos: datos.
  - Tendrá los métodos: Constructor, los métodos get y set de cada uno de los atributos, generar un informe, obtener informe de incidencia por id.
- Archivo
  - Tendrá el atributo: informe.
  - Tendrá los métodos: Constructor, método get y set del atributo informe, descargar archivo como pdf, descargar archivo como xlsx.

Estas clases están vinculadas al subsistema de Exportar.

## Diagrama de Interacción



**Fig. 1.11:** Diagrama de secuencia CU11.

## Caso de Uso 12: Realizar análisis de riesgo

### Especificación de caso de uso de diseño

CASO DE USO N°12	Realizar análisis de riesgo
Actor Principal	Líder de proyecto, Desarrollador
Precondiciones	Existe al menos un riesgo cargado.
Disparador	Un actor añade un riesgo, evaluación o plan de mitigación o minimización y de contingencia, o un proyecto avanza a la siguiente iteración.

DESCRIPCION	Paso	Acción
EXTENSIONES	1	El actor añade un riesgo. Si el actor añade una evaluación continua en extensión 1a. Si el actor añade un plan de contingencia y de minimización o mitigación continua en extensión 1b. Si el proyecto avanza a la siguiente iteración continua en extensión 1c.
	2	El GestorRiesgo solicita a InterfazUsuario marcarRiesgo("Evaluar"). La InterfazUsuario muestra una marca visual del riesgo, notificando que necesita ser evaluado.
	Paso	Acción de Derivación
	1a	Si el factor de riesgo resultado de la evaluación es menor a 9 el GestorRiesgo envía el evento marcarRiesgo("Ignorar") a InterfazUsuario. La InterfazUsuario muestra una marca visual del riesgo, notificando que no necesita acciones y deshabilita las opciones evaluar y planificar. Si el factor de riesgo resultado de la evaluación es mayor igual a 9 y menor que 36 el GestorRiesgo envía el evento marcarRiesgo("Reevaluar") a InterfazUsuario. La InterfazUsuario muestra una marca visual del riesgo, notificando que necesita ser reevaluado en la próxima iteración, y deshabilita las opciones evaluar y planificar. Si el factor de riesgo resultado de la evaluación es mayor o igual a 36 y menor que 64 el GestorRiesgo envía el evento marcarRiesgo("Planificar") a InterfazUsuario. La InterfazUsuario muestra una marca visual del riesgo, notificando que necesita ser planificado, y habilita la opción planificar. Si el factor de riesgo resultado de la evaluación es mayor o igual que 64 el GestorRiesgo envía el evento marcarRiesgo("Critico") a InterfazUsuario. La InterfazUsuario muestra una marca visual del riesgo, notificando que necesita ser planificado de manera urgente, y habilita la opción planificar.
	1b	El GestorRiesgo envía el evento marcarRiesgo("Planificado") a InterfazUsuario. La InterfazUsuario muestra una marca visual del riesgo, notificando que existen planes de riesgo de ese riesgo para la iteración actual.
	1c	El GestorRiesgo envía el evento marcarRiesgos() a InterfazUsuario. La InterfazUsuario muestra una marca visual en todos los riesgos activos (que no estén marcados como "ignorar"), notificando que los riesgos necesitan ser evaluados, habilita la opción de evaluación y deshabilita la de planificación.

## Diagrama de paquetes

Los objetos que fueron identificados en este caso de uso fueron:

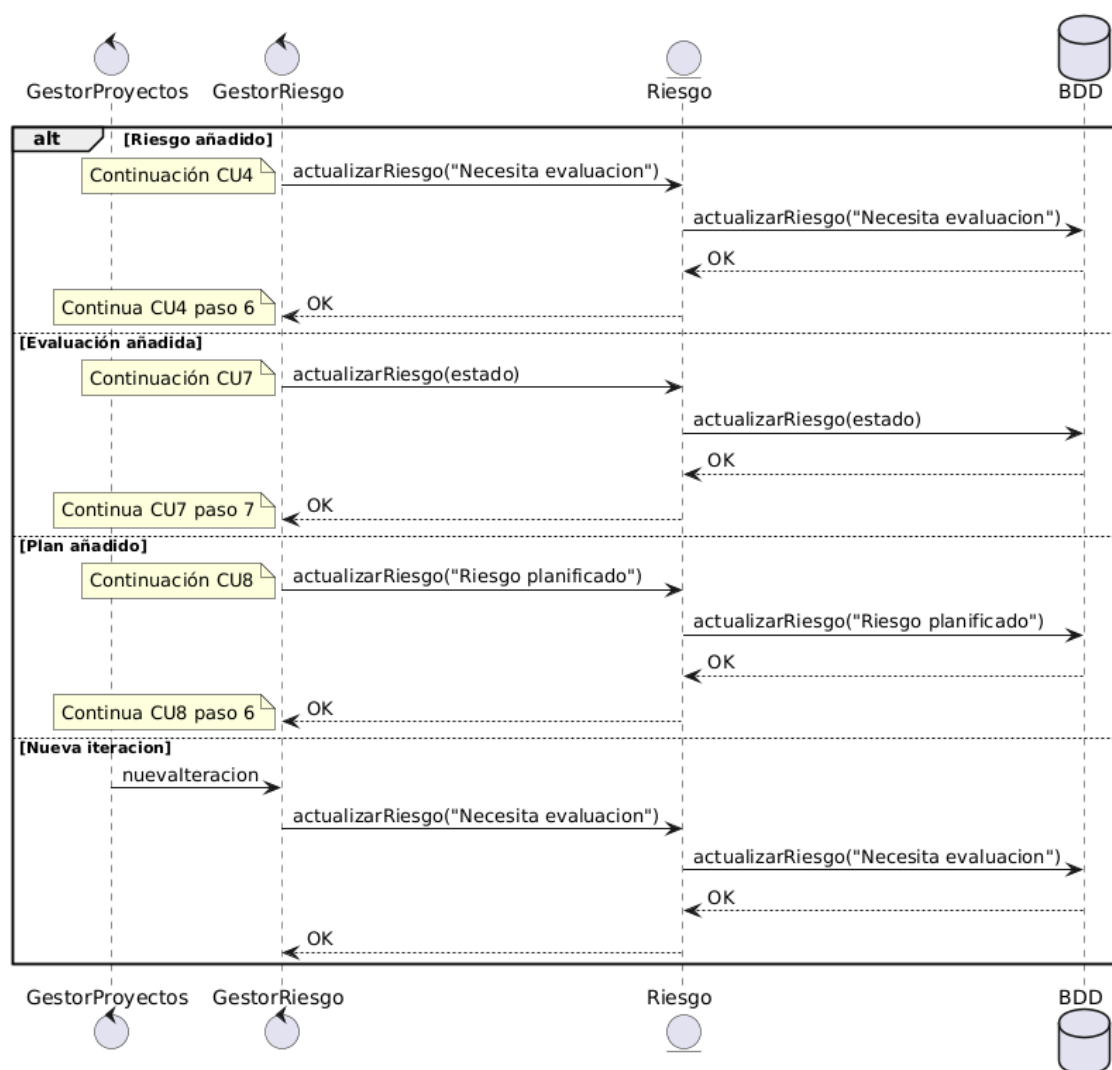
- GestorRiesgo.
  - Tendrá los métodos: Constructor, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, vincular o desvincular responsable, actualizar categoría de un riesgo, vincular o desvincular iteraciones, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id, Crear tarea, modificar tarea por id, vincular o desvincular participante, eliminar tarea por id, Crear plan de riesgo, actualizar

plan de riesgo por id, eliminar plan de riesgo por id, vincular iteración, Vincular o desvincular tareas, crear incidencia, obtener incidencias por id, obtener todas las incidencias, comprobar si riesgo necesita evaluación, comprobar si riesgo necesita un plan, comprobar si riesgo necesita dos planes.

- Riesgo.
  - Tendrá los atributos: descripción, factor riesgo, categoría, iteración y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, agregar riesgo, obtener todos los riesgos de un proyecto, obtener riesgo por id, actualizar riesgo por id, eliminar riesgo por id, actualizar categoría de un riesgo, vincular o desvincular proyecto, vincular o desvincular iteraciones.
- Categoría
  - Tendrá los atributos: nombre, descripción y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Agregar categoría, actualizar categoría, eliminar categoría, obtener todas las categorías.
  - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia.
- Evaluacion
  - Tendrá los atributos: impacto, probabilidad y conexión
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, crear una evaluación, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluación por id, eliminar evaluación por id.
  - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia.
- PlanDeRiesgo
  - Tendrá los atributos: nombre, descripción, tipo, tareas y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Crear plan de riesgo, actualizar plan de riesgo por id, eliminar plan de riesgo por id, vincular iteración, Vincular o desvincular tareas.
- Tareas:
  - Tendrá los atributos: nombre, descripción, estado, fecha inicio, fecha fin y conexión.
  - Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, Crear tarea, modificar tarea por id, vincular o desvincular participante, eliminar tarea por id.
- Incidencia:
  - Tendrá los atributos: descripción y conexión
  - Tendrá los métodos: Tendrá los métodos: Constructor, los métodos get y set de cada atributo excepto para conexión, crear incidencia, obtener incidencias por id de riesgo, obtener todas las incidencias.
  - Comentario: Es un tipo de informe.

Estas clases están vinculadas al subsistema de Riesgos.

## Diagrama de Interacción

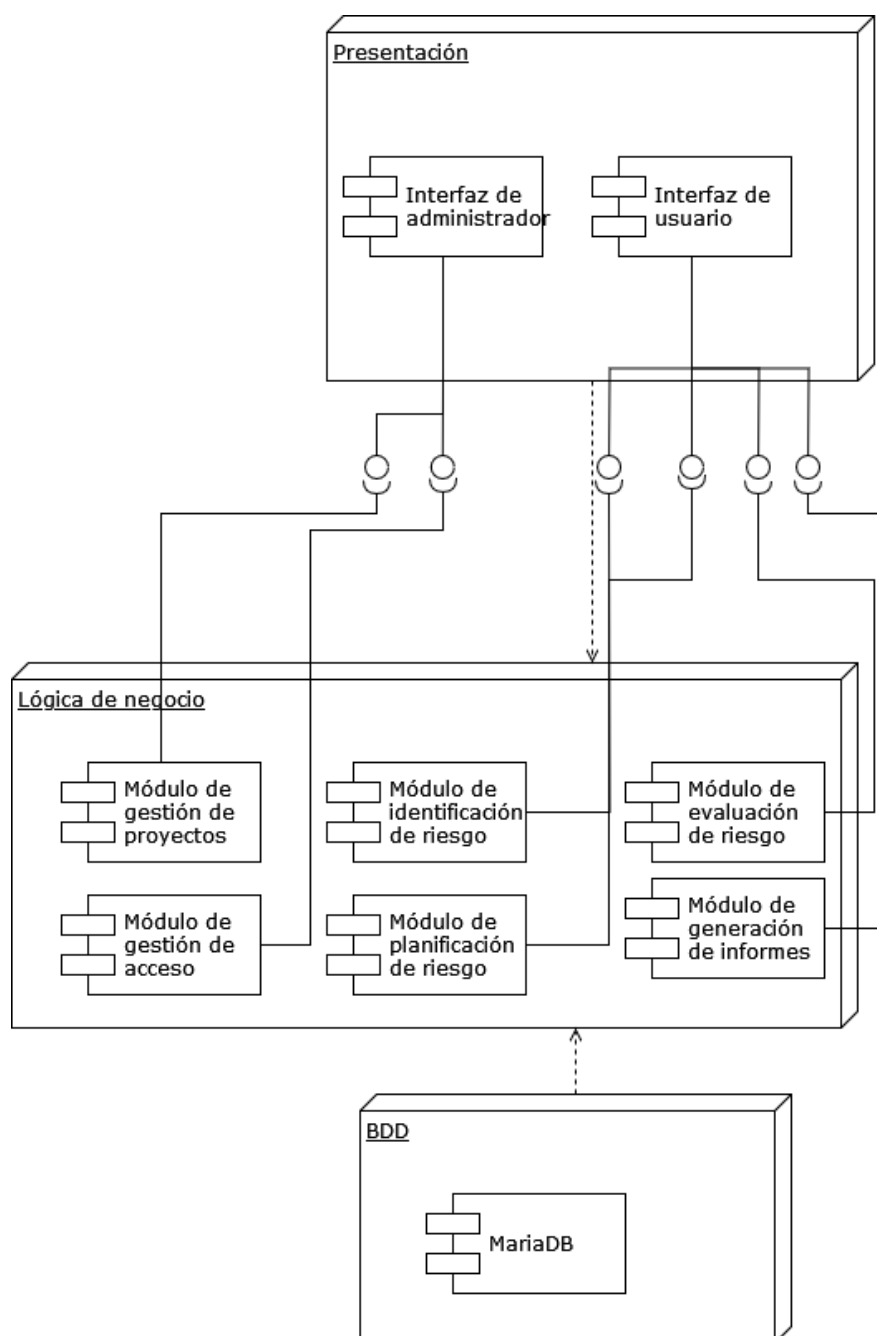


**Fig. 1.12:** Diagrama de secuencia CU12.



## Diagramas

### Diagrama de componentes



**Fig. 2.1:** Modelo de componentes.

## Diagrama de Clases

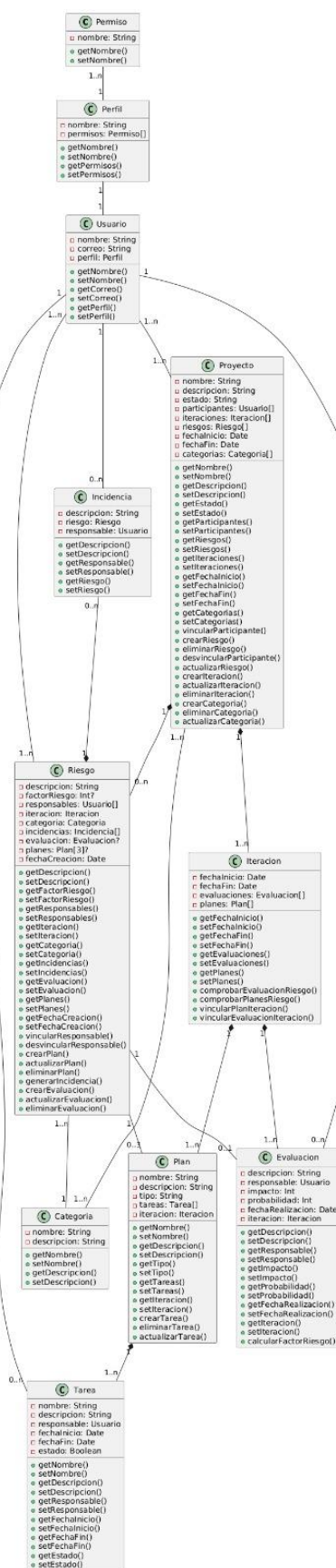
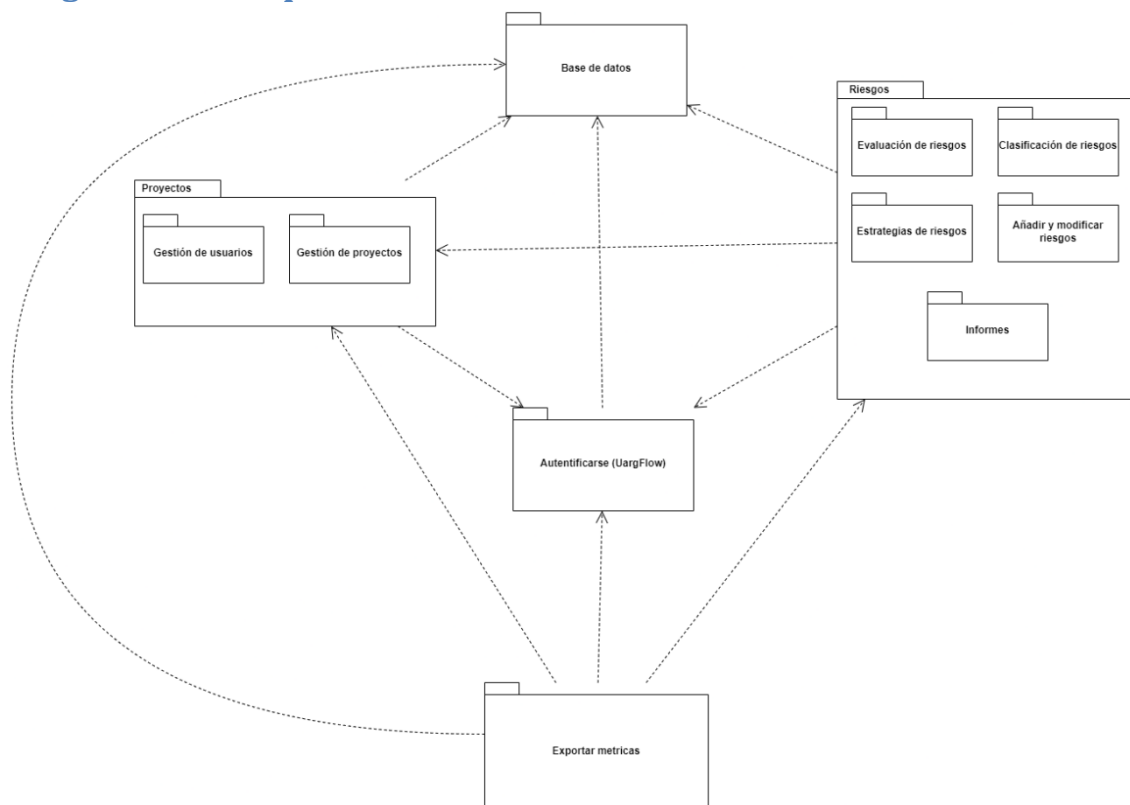


Fig. 2.2: Diagrama de Clases.

**Aclaracion:** Este diagrama esta pendiente a modificaciones.

## Diagramas de Paquetes



**Fig. 2.3:** Diagrama de Paquetes.