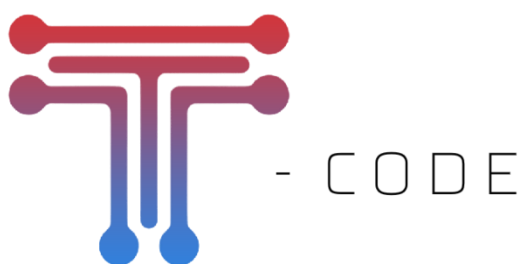


Modelo de Diseño

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

Introducción	5
<i>Propósito</i>	<i>5</i>
<i>Alcance</i>	<i>5</i>
Referencias.....	5
<i>Visión general.....</i>	<i>6</i>
Diseño de Casos de Uso	6
<i>Caso de Uso 1: Autenticarse.....</i>	<i>6</i>
Diagrama de paquetes.....	6
Diagrama de Interacción.....	6
Requerimientos especiales o de implementación	7
<i>Caso de Uso 2: Administrar acceso al sistema</i>	<i>7</i>
Diagrama de paquetes.....	7
Diagrama de Interacción.....	8
<i>Caso de Uso 3: Administrar proyectos</i>	<i>10</i>
Diagrama de paquetes.....	10
Diagrama de Interacción.....	11
<i>Caso de Uso 4: Añadir riesgo a la lista</i>	<i>13</i>
Diagrama de paquetes.....	13
Diagrama de Interacción.....	14
<i>Caso de Uso 5: Modificar lista de riesgos.....</i>	<i>15</i>
Diagrama de paquetes.....	15
Diagrama de Interacción.....	17
<i>Caso de Uso 6: Administrar categorías de riesgos</i>	<i>18</i>
Diagrama de paquetes.....	18
Diagrama de Interacción.....	19
<i>Caso de Uso 7: Realizar evaluación de riesgo</i>	<i>20</i>
Diagrama de paquetes.....	20
Diagrama de Interacción.....	21
<i>Caso de Uso 8: Añadir plan de acción</i>	<i>21</i>
Diagrama de paquetes.....	21
Diagrama de Interacción.....	23
<i>Caso de Uso 9: Modificar plan de acción.....</i>	<i>23</i>
Diagrama de paquetes.....	23
Diagrama de Interacción.....	25
<i>Caso de Uso 10: Realizar informes</i>	<i>25</i>
Diagrama de paquetes.....	25
Diagrama de Interacción.....	28

<i>Caso de Uso 11: Exportar archivos</i>	30
Diagrama de paquetes	30
Diagrama de Interacción	31
<i>Caso de Uso 12: Realizar análisis de riesgo</i>	31
Diagrama de paquetes	31
Estas clases están vinculadas al subsistema de Riesgos. Diagrama de Interacción	33
Diagramas	34
<i>Diagrama de componentes</i>	34
<i>Diagrama de Clases</i>	35
<i>Diagramas de Paquetes</i>	36

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

Diagrama de paquetes

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

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

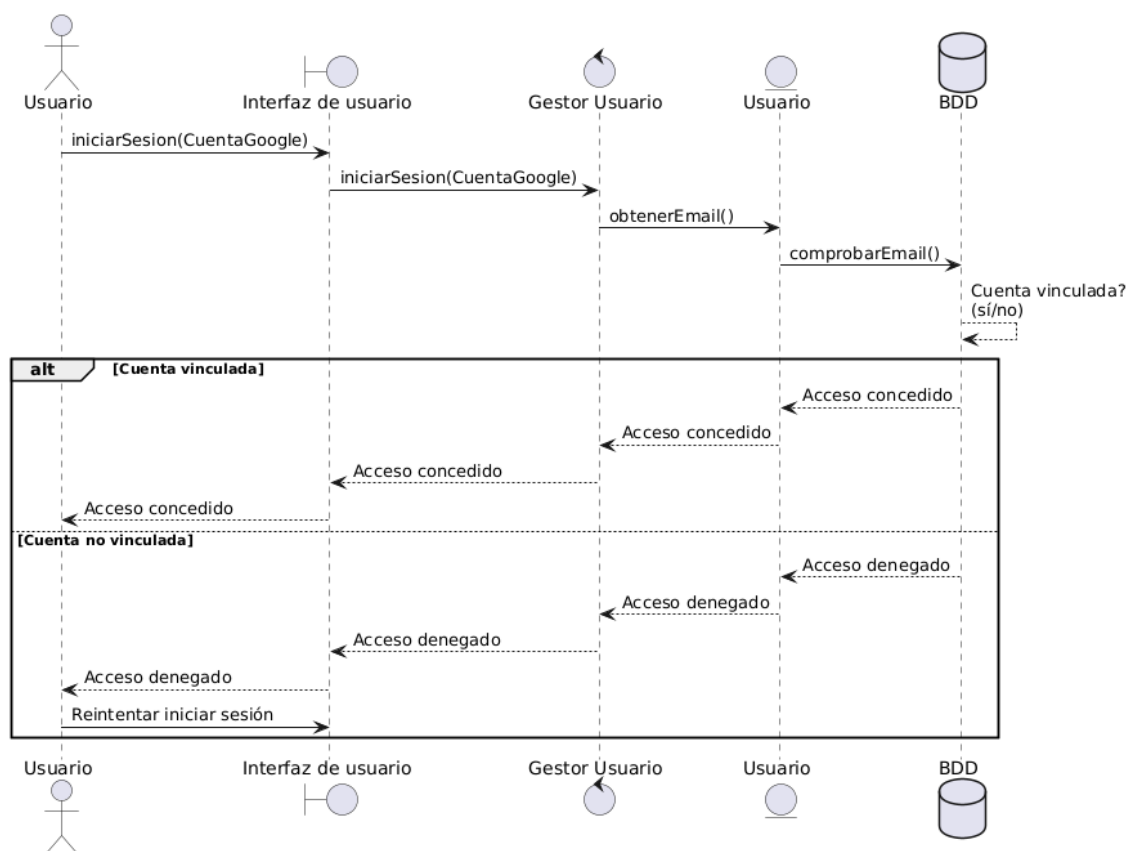


Fig. 1.1: Diagrama de secuencia CU01.

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

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

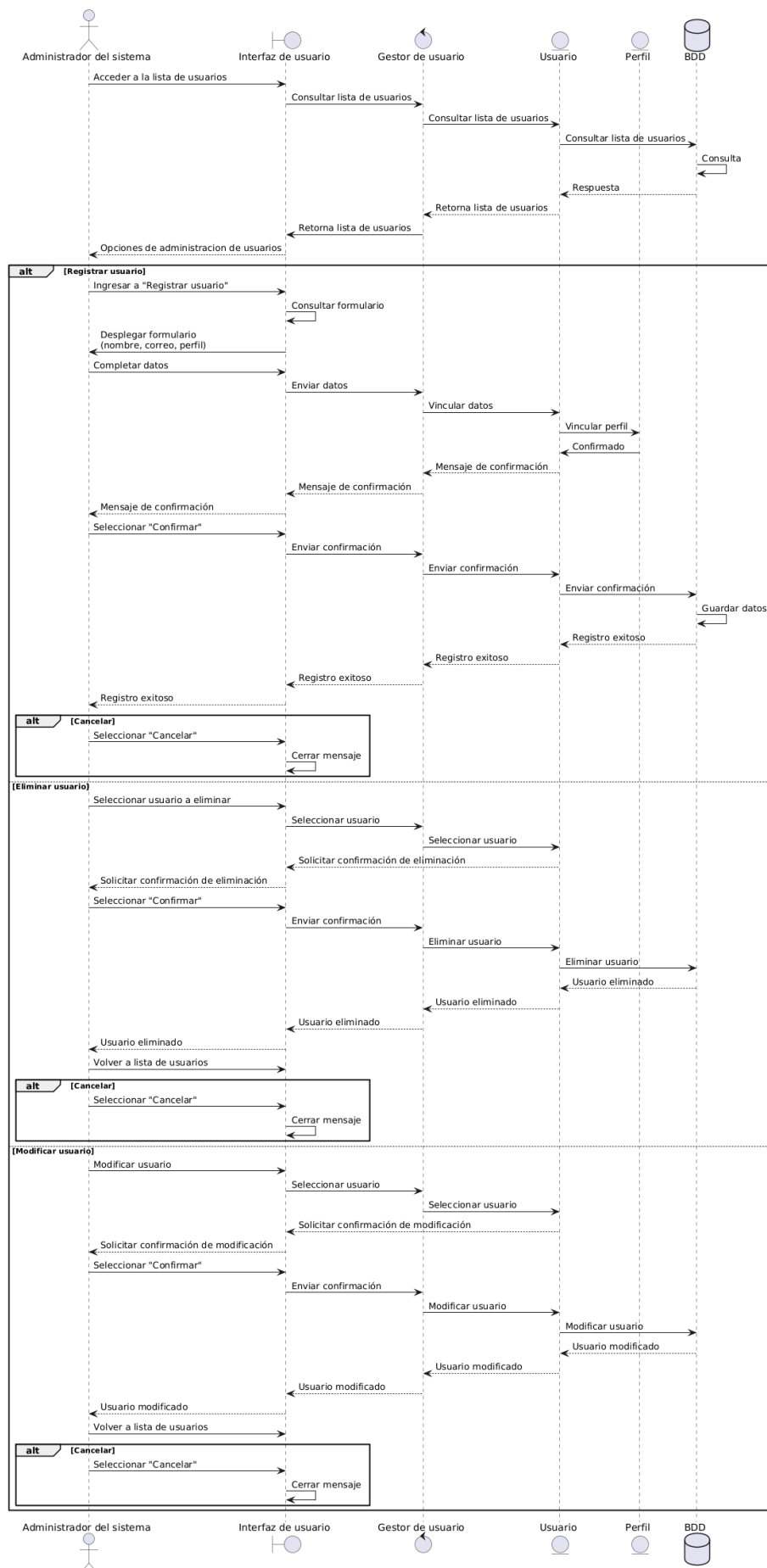


Fig. 1.2: Diagrama de secuencia CU02.

Caso de Uso 3: Administrar proyectos

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 iteracion por id, eliminar iteracion por id, añadir iteracion.
 - Comentario: Cada método deberá comprobar que no se pase ningún campo vacío, ni ninguna inyección sql.
- Proyecto.
 - Tendrá los atributos de nombre, descripcion, 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 iteracion por id, eliminar iteracion por id, añadir iteracion.

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

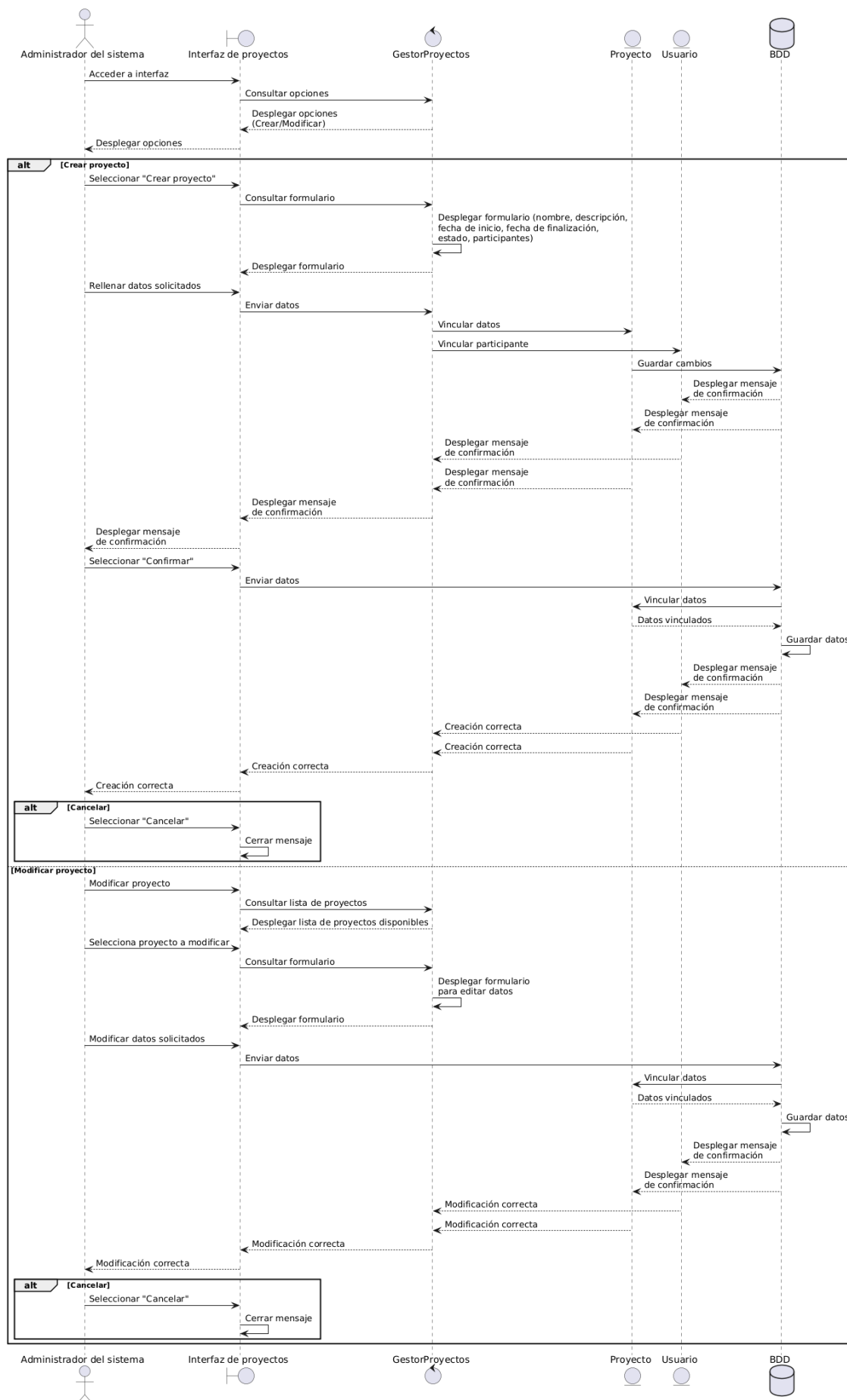


Fig. 1.3: Diagrama de secuencia CU03.

Caso de Uso 4: Añadir riesgo a la lista

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: descripcion, factor_riesgo, categoría, iteracion 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, descripcion 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 iteracion por id, eliminar iteracion por id, añadir iteracion.
 - Comentario: Cada método deberá comprobar que no se pase ningún campo vacío, ni ninguna inyección sql.
- Proyecto.
 - Tendrá los atributos de nombre, descripcion, 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 iteracion por id, eliminar iteracion por id, añadir iteracion.

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

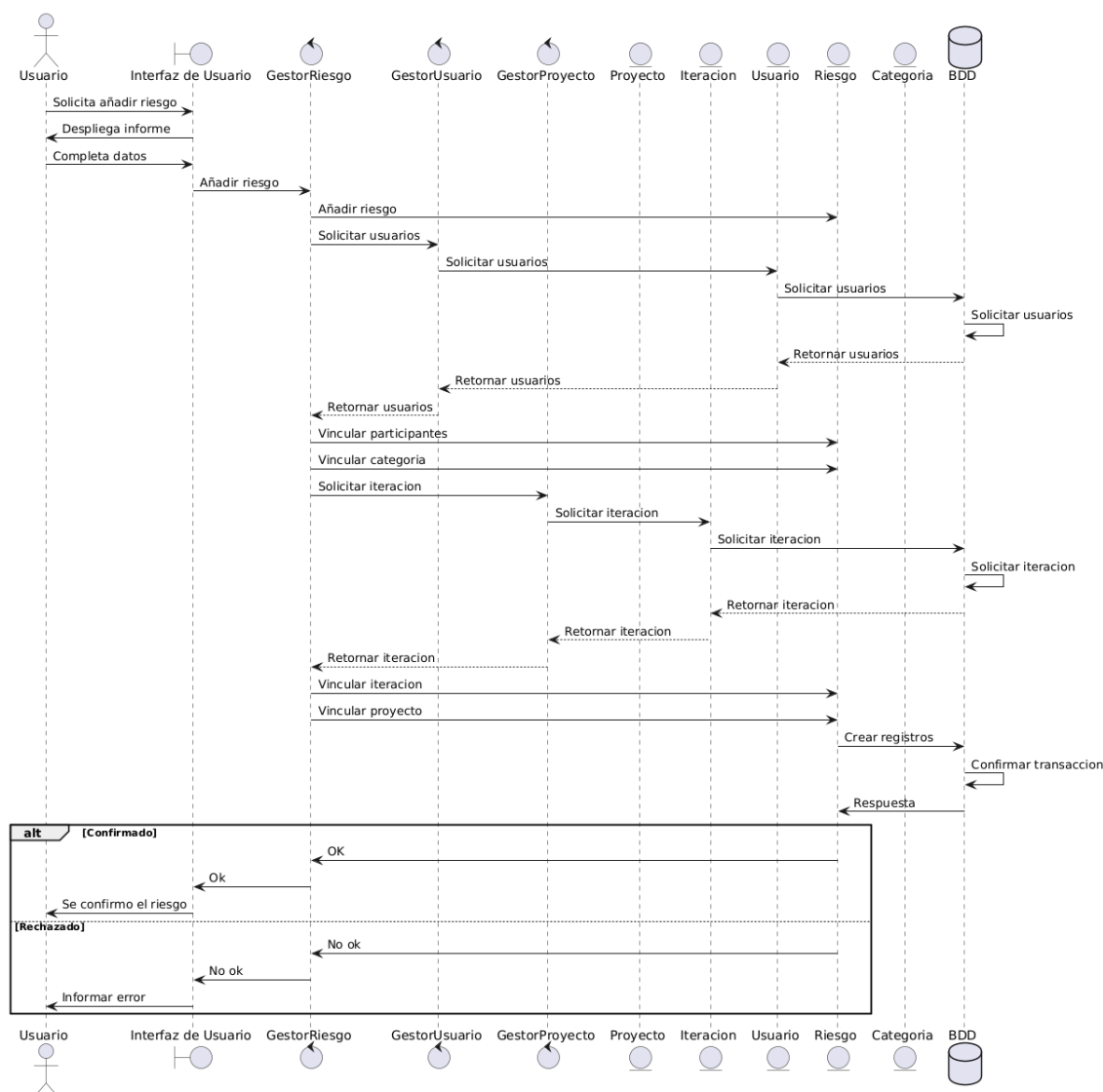


Fig. 1.4: Diagrama de secuencia CU04.

Caso de Uso 5: Modificar lista de riesgos

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 evaluacion, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluacion por id, eliminar evaluacion por id.
- Riesgo.
 - Tendrá los atributos: descripcion, factor_riesgo, categoría, iteracion 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, descripcion 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 evaluacion, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluacion por id, eliminar evaluacion 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 iteracion por id, eliminar iteracion por id, añadir iteracion.

- 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.
- Iteración.
 - 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

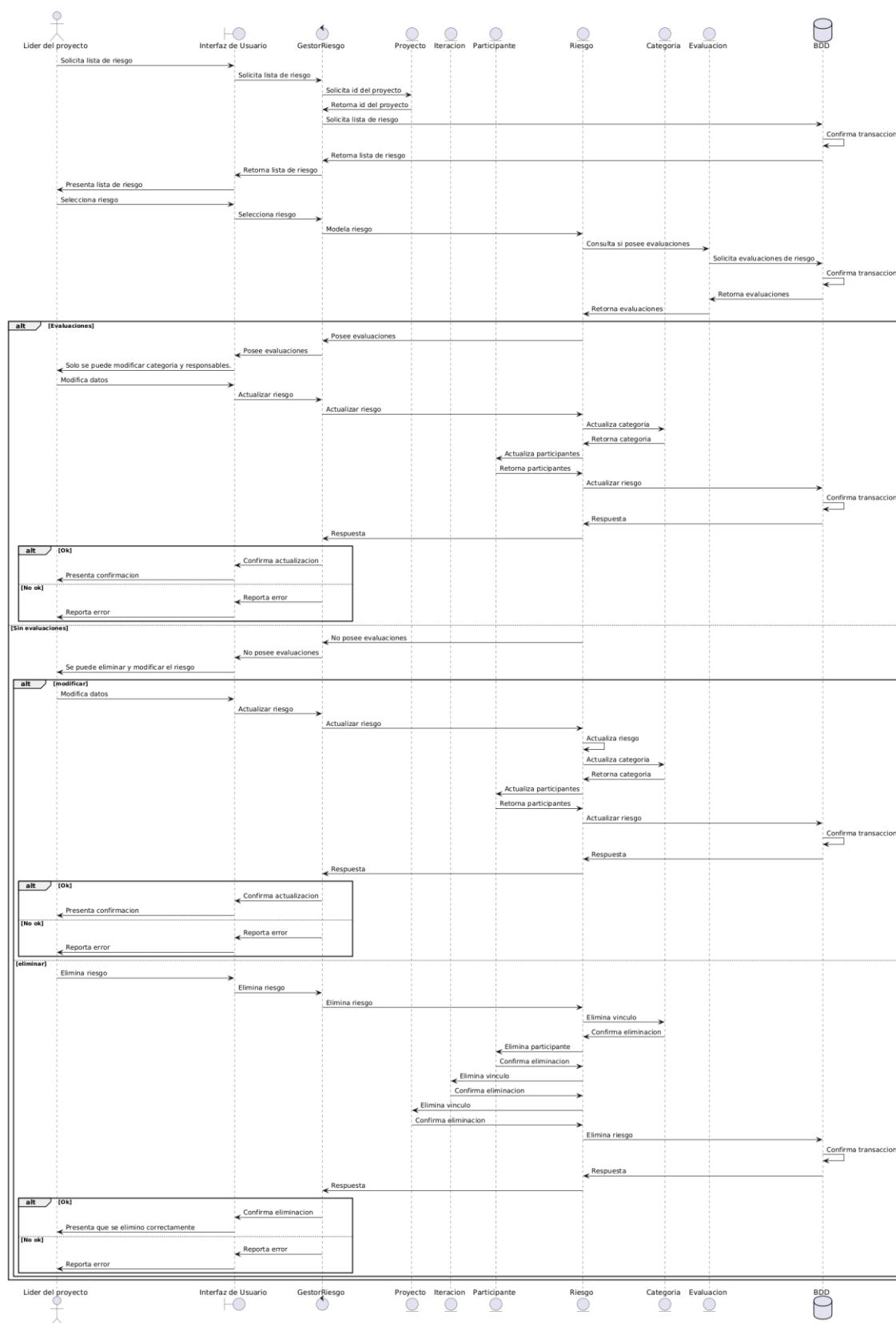


Fig. 1.5: Diagrama de secuencia CU05.

Caso de Uso 6: Administrar categorías de riesgos

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: descripcion, factor_riesgo, categoría, iteracion 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, descripcion 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.

Estas clases están vinculadas al subsistema de Riesgos.

Diagrama de Interacción

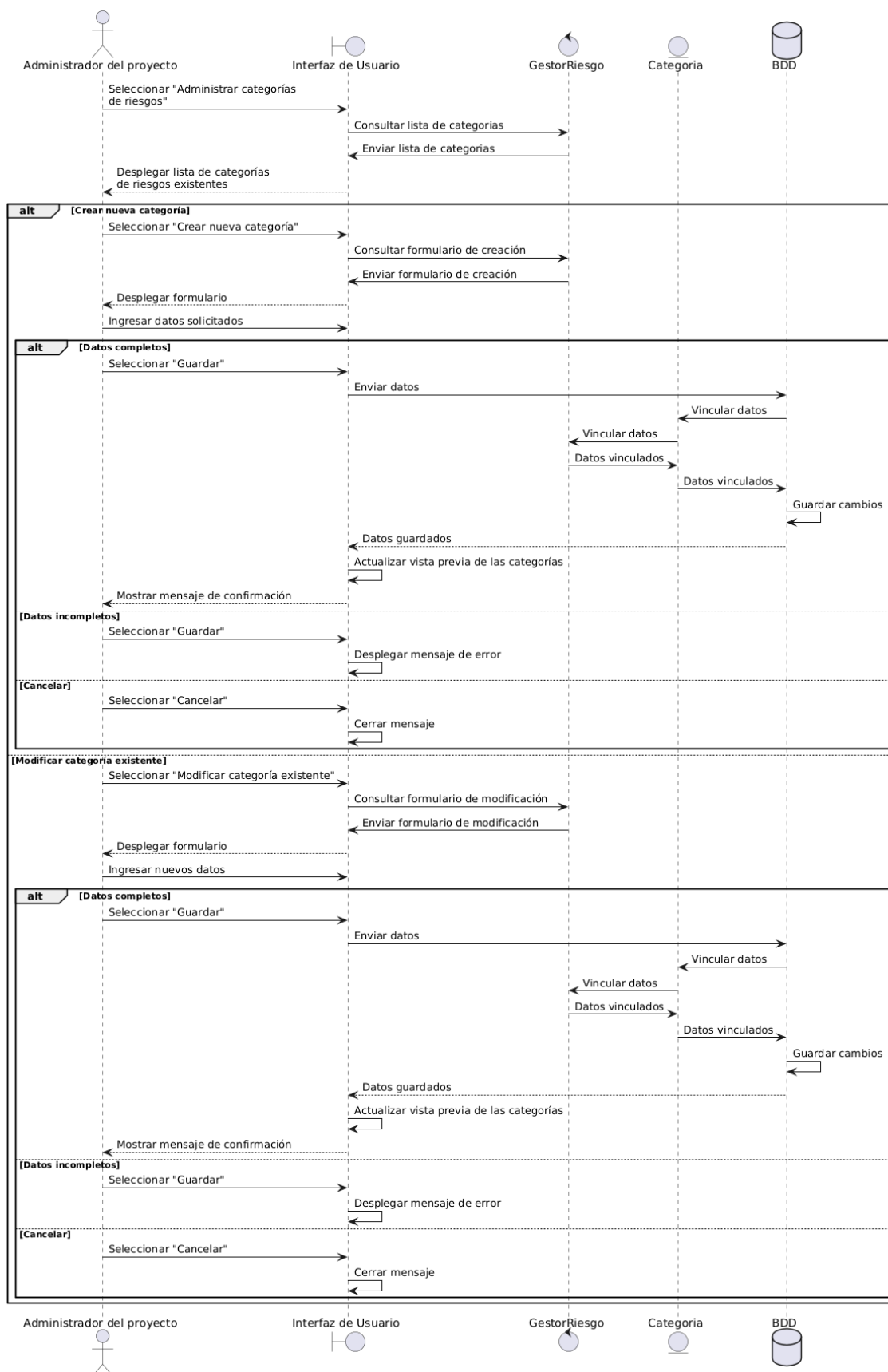


Fig. 1.6 Diagrama de secuencia CU06.

Caso de Uso 7: Realizar evaluación de riesgo

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 evaluacion, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluacion por id, eliminar evaluacion por id.
- Riesgo.
 - Tendrá los atributos: descripcion, factor_riesgo, categoría, iteracion 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, descripcion 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 evaluacion, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluacion por id, eliminar evaluacion por id.

Estas clases están vinculadas al subsistema de Riesgos.

Diagrama de Interacción

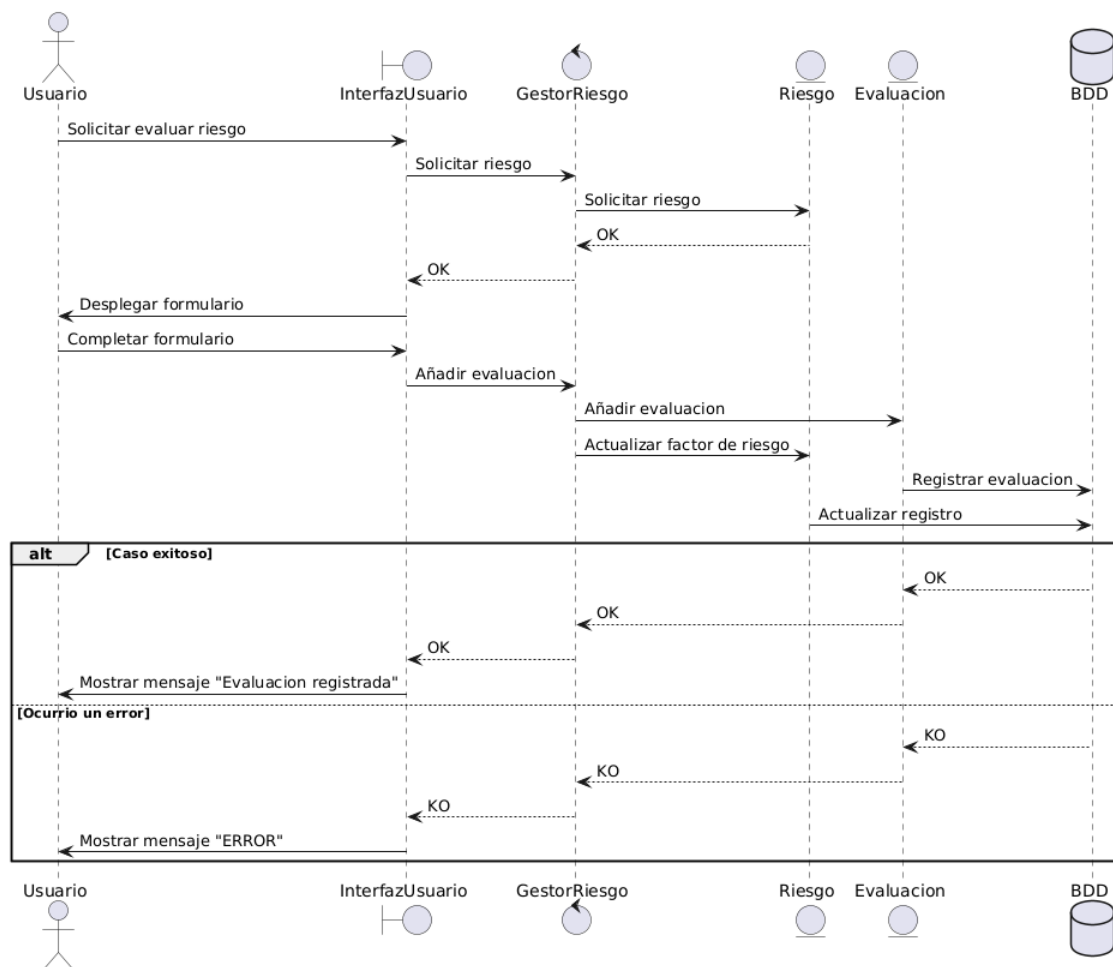


Fig. 1.7: Diagrama de secuencia CU07.

Caso de Uso 8: Añadir plan de acción

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: descripcion, factor_riesgo, categoría, iteracion 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 evaluacion, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluacion por id, eliminar evaluacion 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

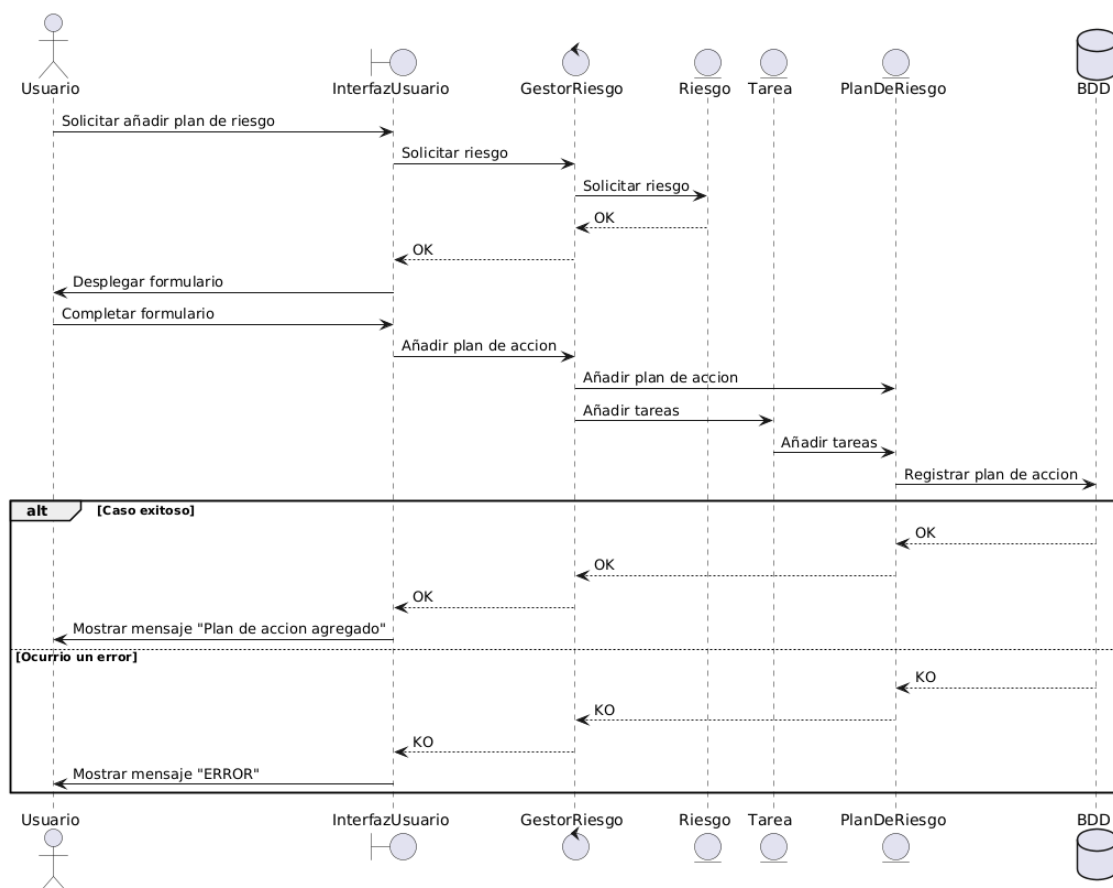


Fig. 1.8: Diagrama de secuencia CU08.

Caso de Uso 9: Modificar plan de acción

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.
- Riesgo.
 - Tendrá los atributos: descripcion, factor_riesgo, categoría, iteracion 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.
- Evaluación
 - 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

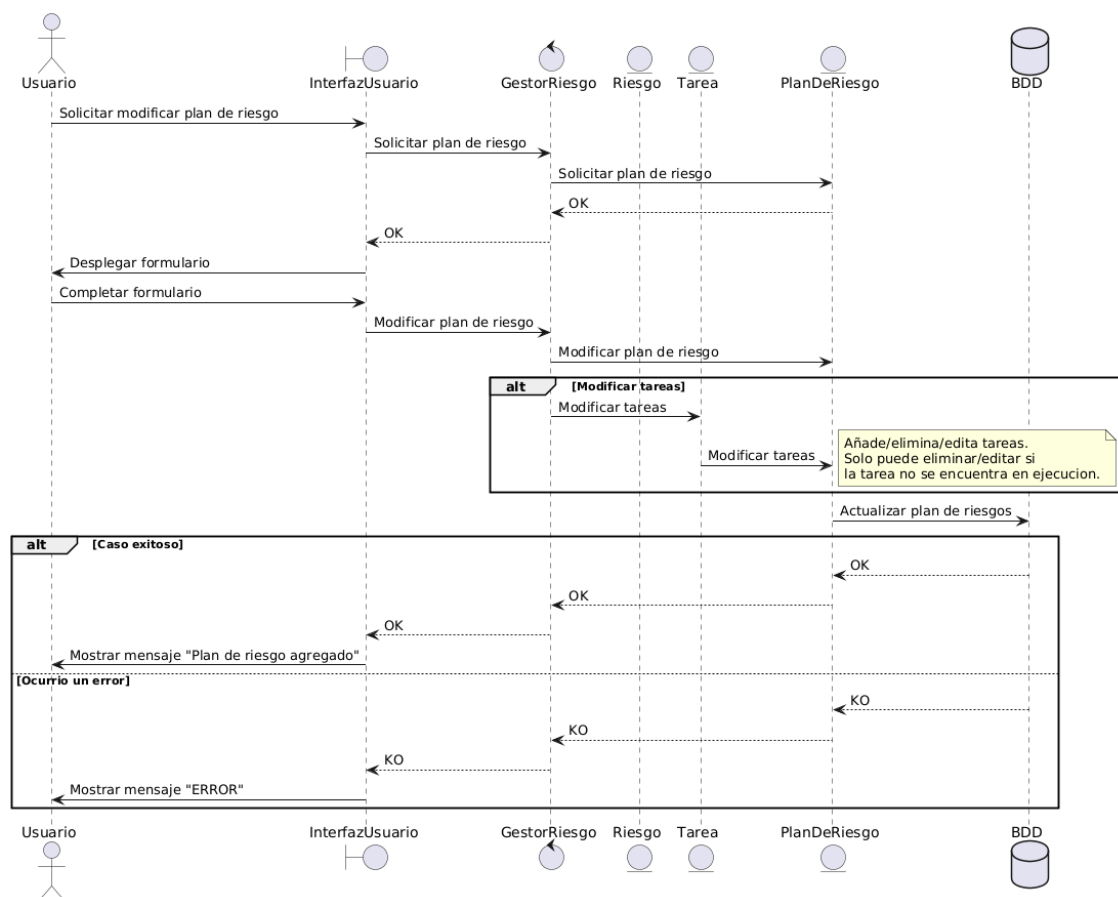


Fig. 1.9: Diagrama de secuencia CU09.

Caso de Uso 10: Realizar informes

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 evaluacion, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluacion por id, eliminar evaluacion 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: descripcion, factor_riesgo, categoría, iteracion 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.
- Evaluación
 - 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 xslx.
- 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

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

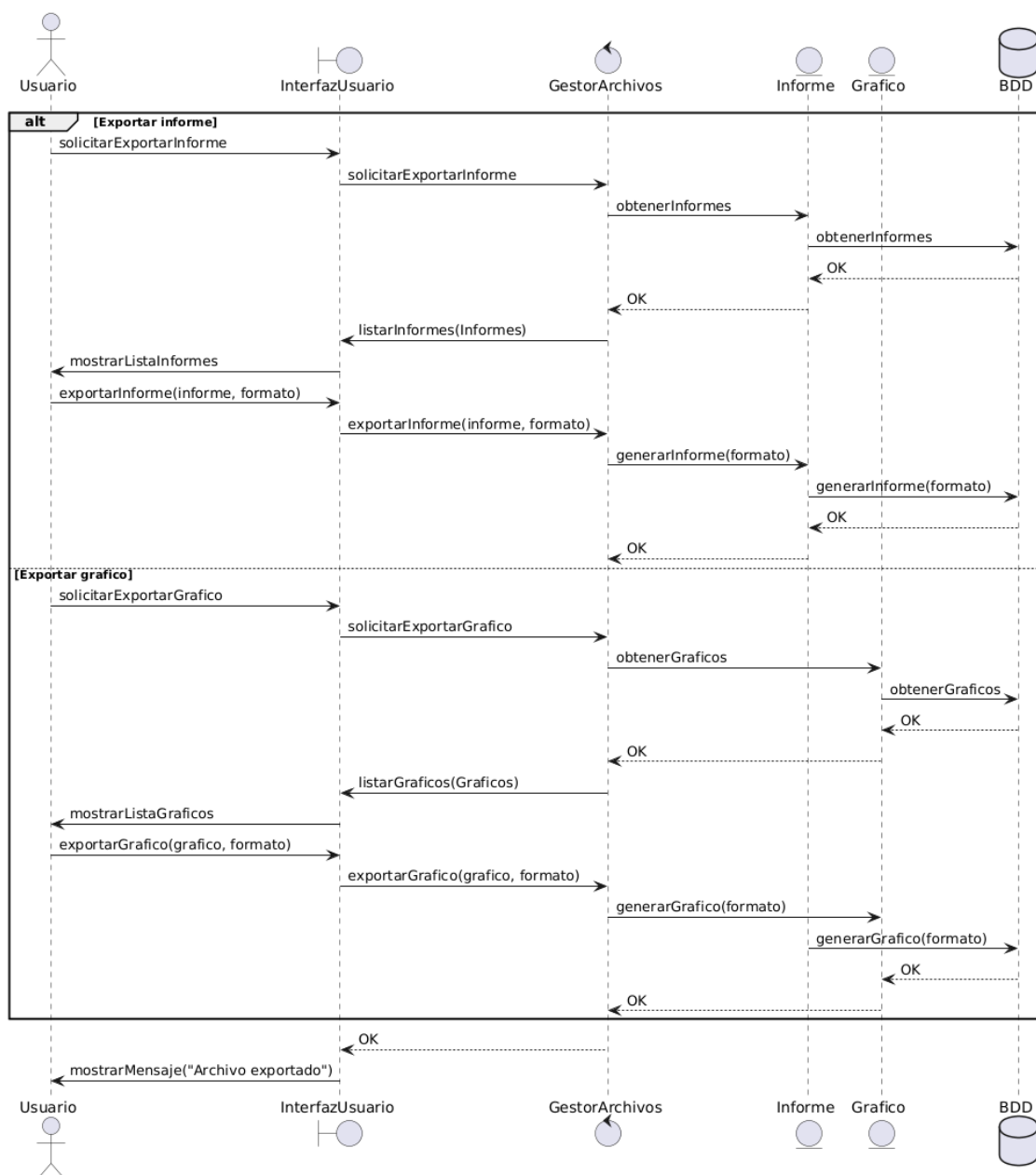


Fig. 1.11: Diagrama de secuencia CU11.

Caso de Uso 12: Realizar análisis de riesgo

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 evaluacion, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluacion por id, eliminar evaluacion 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 evaluacion, comprobar si riesgo necesita un plan, comprobar si riesgo necesita dos planes.

- Riesgo.
 - Tendrá los atributos: descripcion, factor_riesgo, categoría, iteracion 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, descripcion 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 evaluacion, obtener evaluaciones de un riesgo, obtener todas las evaluaciones, actualizar evaluacion por id, eliminar evaluacion por id.
 - Comentario: Esta clase se utiliza de manera implícita en este diagrama de secuencia
- PlanDeRiesgo
 - Tendrá los atributos: nombre, descripcion, 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, descripcion, 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

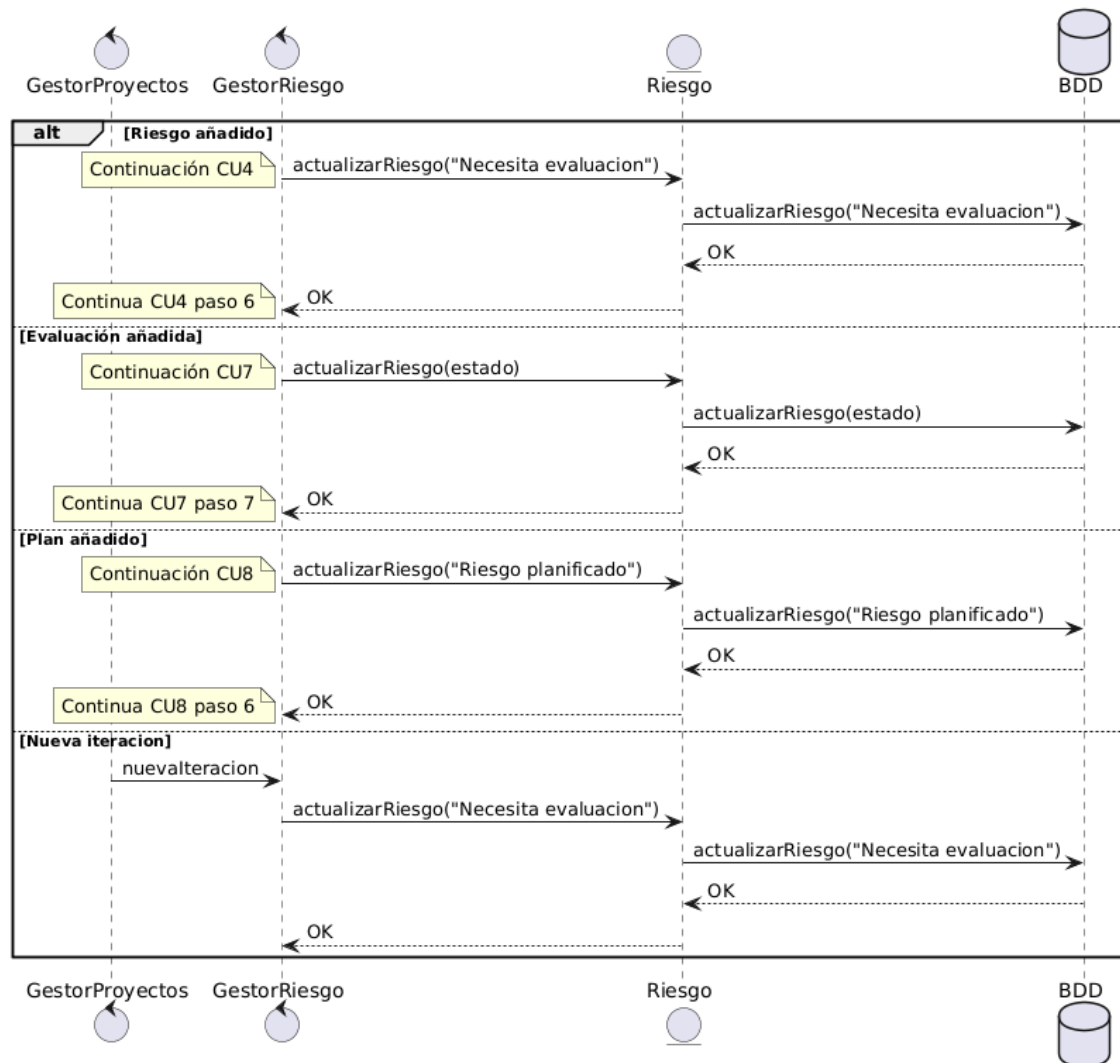


Fig. 1.12: Diagrama de secuencia CU12.

Diagramas

Diagrama de componentes

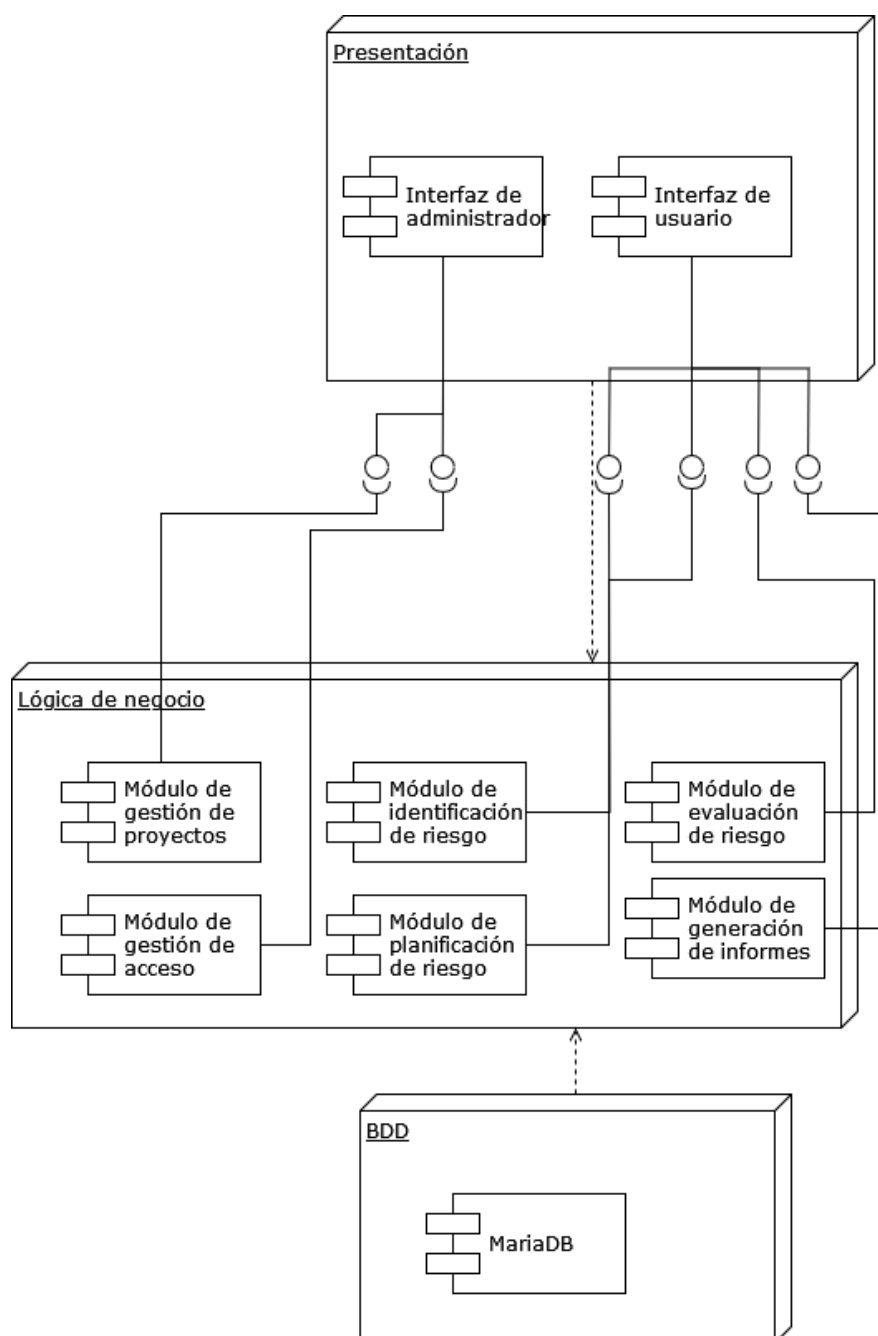


Fig. 2.1: Modelo de componentes.

Diagrama de Clases

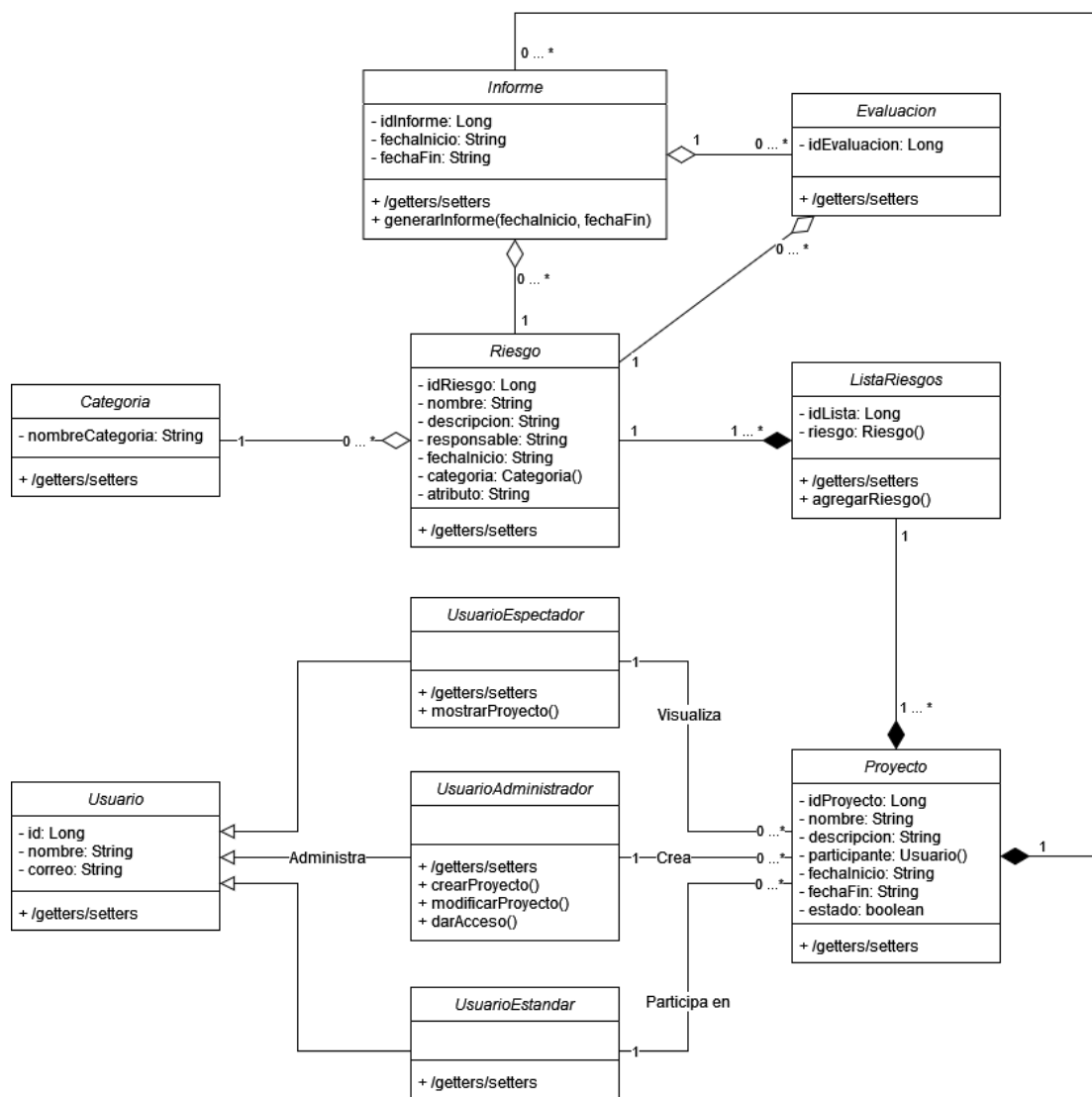


Fig. 2.2: Diagrama de Clases.

Aclaracion: Este diagrama esta pendiente a modificaciones.

Diagramas de Paquetes

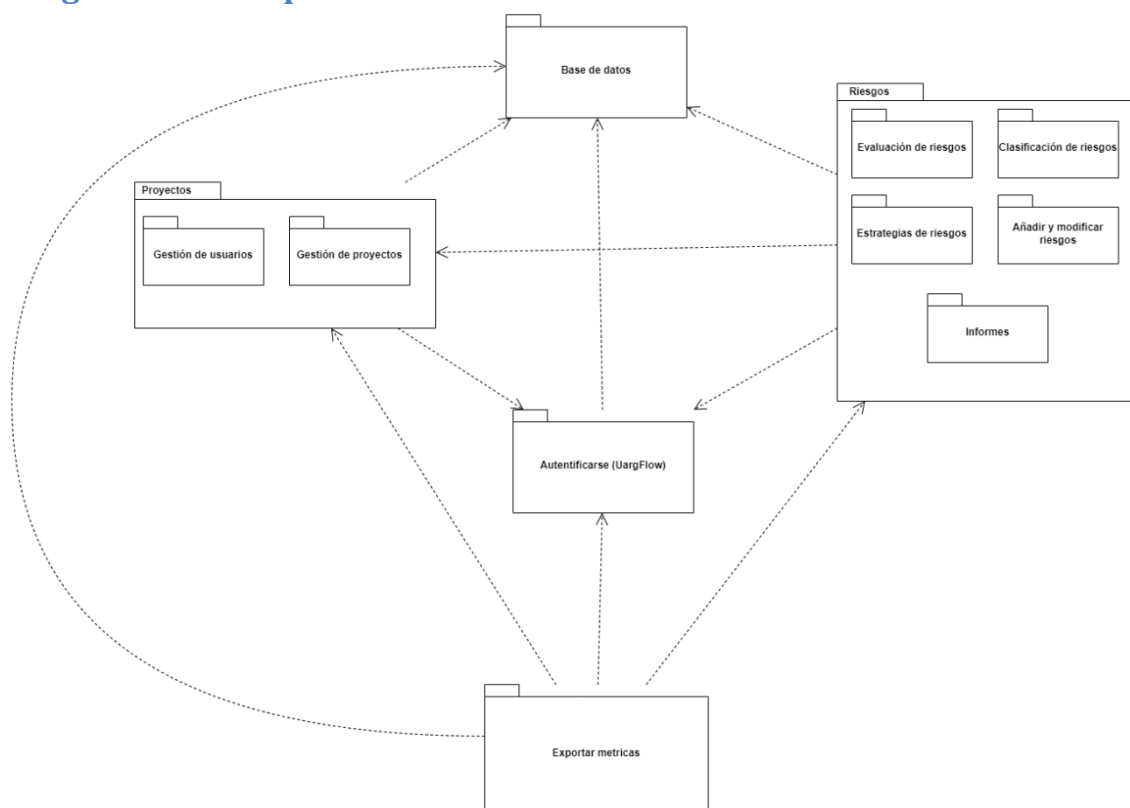


Fig. 2.3: Diagrama de Paquetes.