**Proyecto SAR**

**Documento de Diseño**

EC-DD. Versión 3.0

Fecha: 19/04/2018

Estatus: Restringido

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Versión** | **Fecha** | **Descripción del Elemento / Versión** | **Realización** | **Validación** | **Revisión Formal** |
| EC-DD | 1.0 | 12/03/2018 | Documento de diseño a nivel de arquitectura del sistema. | Equipo de Desarrollo | Equipo de Desarrollo | Jefe de proyecto |
| EC-DD | 2.0 | 12/04/2018 | Modificación del diagrama de clases para contemplar el captcha. | Equipo de Operación | Autoridad de Configuración | Javier Gómez Martínez, responsable de revisiones |
| EC-DD | 3.0 | 19/04/2018 | Modificación del diagrama de clases para contemplar los reportes avanzados. | Equipo de Operación | Autoridad de Configuración | Javier Gómez Martínez, responsable de revisiones |

**Resumen**

En este documento vamos a detallar los aspectos relativos al diseño del Sistema de Administración de Requisitos.

El documento está dividido en cuatro apartados principales.

* Empezamos con una descripción de la arquitectura del sistema que vamos a diseñar.
* A continuación mostraremos y describiremos el diagrama de clases asociado a nuestro proyecto.
* Seguiremos mostrando los diagramas de secuencia asociados.
* Acabaremos con un apartado de glosario donde definiremos ciertos términos menos intuitivos que usaremos en el documento.

**Índice de Contenidos**

**1. Descripción de la Arquitectura del Sistema 4**

**2. Diagrama de Clases 5**

**3. Diagramas de Secuencia 6**

**3.1 Diagrama de Secuencia 1 – Especificar Requisito 6**

**3.2 Diagrama de Secuencia 3 – Crear Proyecto 6**

**3.3 Diagrama de Secuencia 5 – Crear Solicitud de Requisito 7**

**3.4 Diagrama de Secuencia 7 – Crear Artefacto 9**

**3.5 Diagrama de secuencia 14 - Listar Artefactos 9**

**4. Glosario 10**

**1. Descripción de la Arquitectura del Sistema**

La arquitectura de este sistema informático se basa en un patrón MVC (Modelo Vista Controlador).El principal objetivo de este patrón es dividir los datos del sistema de la presentación de los mismos al usuario. Para ello se utilizan tres componentes básicas:

* Modelo. Está constituido por todos los datos del sistema. Es el encargado de modificarlos, crearlos, destruirlos y gestionarlos.
* Vista. Es el encargado de presentar los datos del modelo al usuario.
* Controlador. Es la componente con la que el usuario interacciona mediante eventos. El controlador responde a dichos eventos obteniendo la información necesaria del modelo y pasando los datos a la vista.

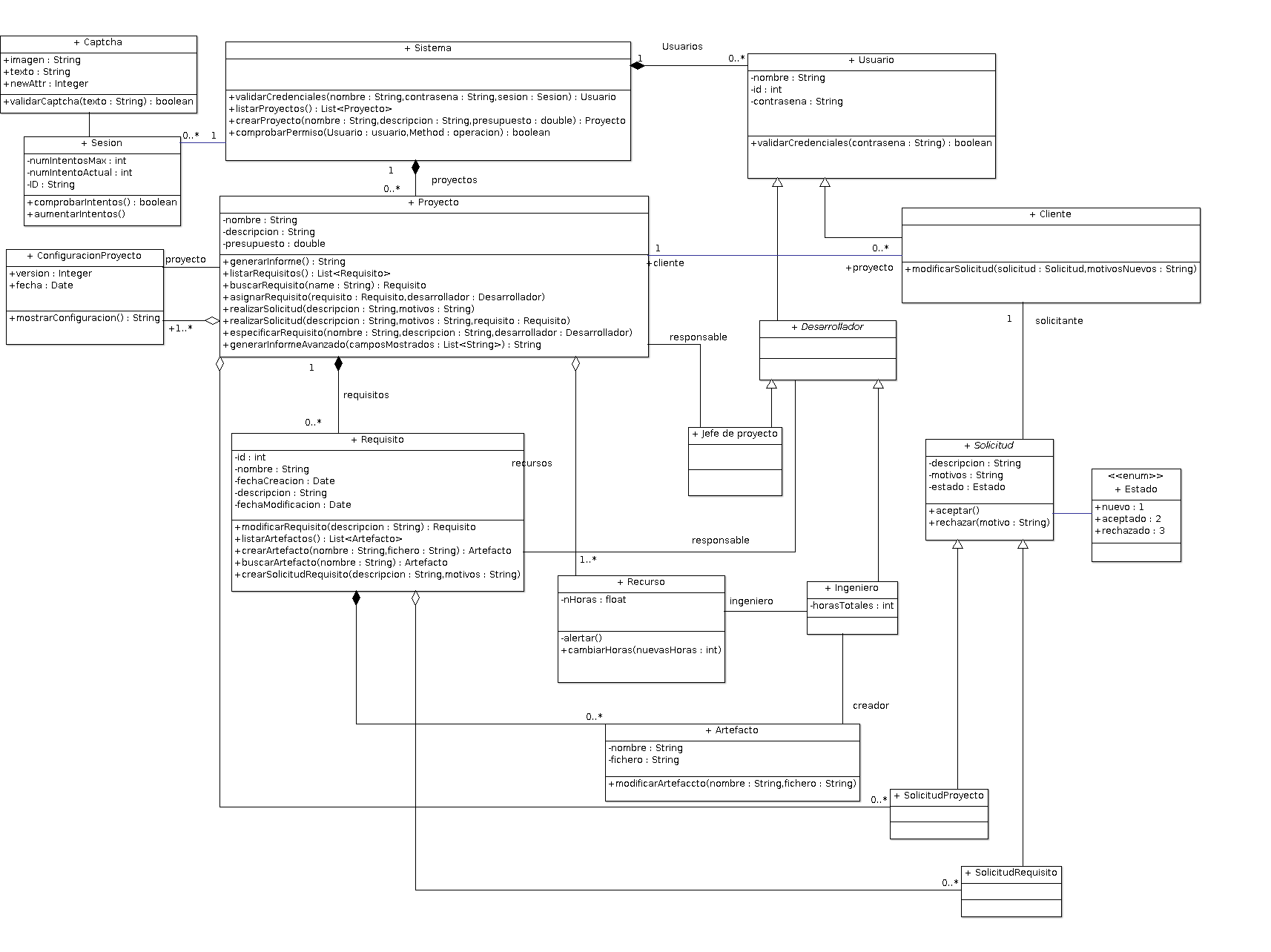
A continuación procedemos a detallar las clases resultantes tras el proceso de diseño del sistema:

* Sistema: Es el encargado de gestionar el resto de clases de la aplicación. Sigue un patrón *singleton.* Contiene y gestiona tanto a los usuarios de la aplicación como a los proyectos.
* Sesión: Representa la sesión que se crea en el servidor cuando un navegador se conecta. Sus atributos se destinan a controlar el número de intentos de un usuario por ingresar en el sistema.
* Captcha: Es la clase encargada de establecer una correspondencia entre imágenes y cadenas de caracteres que representan la solución del Captcha.
* Usuario: Clase abstracta que modela los usuarios de la aplicación, con el objetivo de gestionar sus permisos. De ella heredan las siguientes dos clases:
  + Cliente: Representa a los clientes de la empresa que usará el SAR.
  + Desarrollador: De nuevo una clase abstracta encargada de modelar a los usuarios que desempeñan un rol en la empresa. Un desarrollador puede ser:
    - Ingeniero
    - Jefe de proyecto
* Proyecto: Es la encargada de realizar el objetivo principal de la aplicación. Todo proyecto tendrá un único responsable, que será un jefe de proyecto y un único cliente. Asimismo el proyecto es el encargado de gestionar sus propios requisitos y llevará un registro de todos los ingenieros que estén trabajando en él.
* ConfiguracionProyecto: Clase dedicada a recoger el estado del proyecto en un momento determinado. El principal objetivo es poder mantener el historial de versiones de un proyecto.
* Recurso: Representa a un ingeniero trabajando en un proyecto. Su objetivo es llevar el recuento de horas semanales totales que un ingeniero tiene asignadas.
* Requisito: Siempre forman parte de un proyecto. Pueden contener artefactos y solicitudes de cambio por parte de algún cliente.
* Artefacto: Puesto que los artefactos pueden ser muy diversos, hemos decidido considerarlos como un documento para simplificar la implementación.
* Solicitud: Como en el sistema se han considerado dos tipos de solicitudes hemos decidido que sea una clase abstracta. Sus clases hijas son:
  + SolicitudProyecto: Es una solicitud de carácter general asociada a un proyecto.
  + SolicitudRequisito: Modela una solicitud para modificar un requisito-

Solo se ha considerado una enumeración en el diagrama, Estado, que se encarga de registrar si una solicitud ha sido aceptada, rechazada o si aún está en proceso de valoración.

**2. Diagrama de Clases**

Derivado de la arquitectura explicada en el apartado anterior se obtiene el diagrama de clases mostrado en la figura 01.



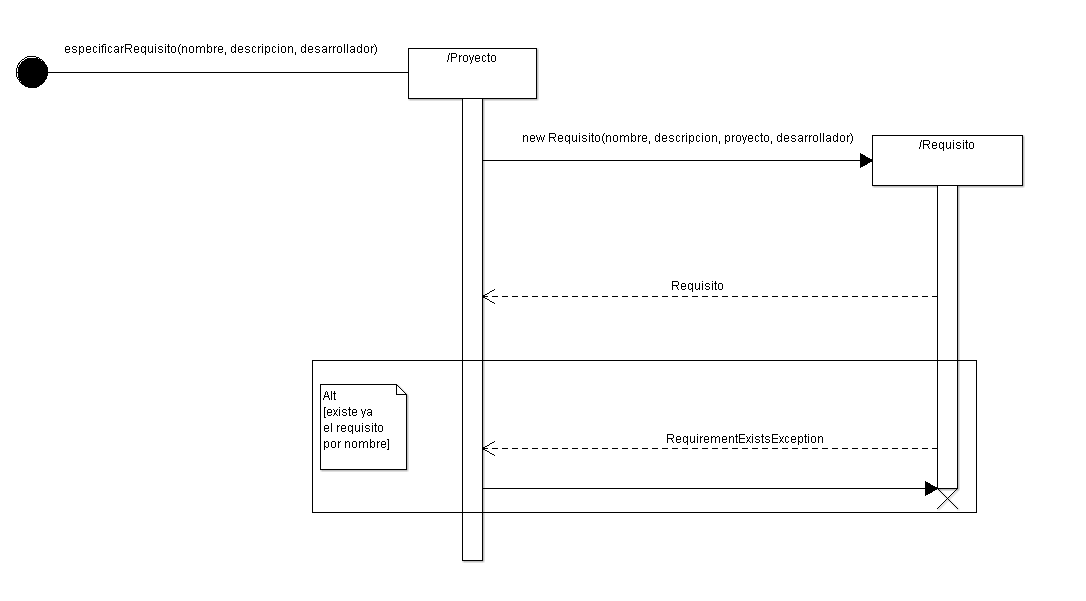
**Figura 01: Diagrama de clases del sistema**

# 3. Diagramas de Secuencia

A lo largo de los diferentes apartados de esta sección se van a detallar los diagramas de secuencia de las diferentes funciones que realizará el sistema.

* 1. **Diagrama de Secuencia 1 – Especificar Requisito**

En la figura 02 podemos ver el proceso que se sigue en el sistema para crear un requisito. Se invoca el método especificarRequisito. Si el sistema detecta que ya existe un requisito con ese mismo nombre, se lanzará una excepción deteniendo el proceso.



**Figura 02: Diagrama secuencia Especificar Requisito**

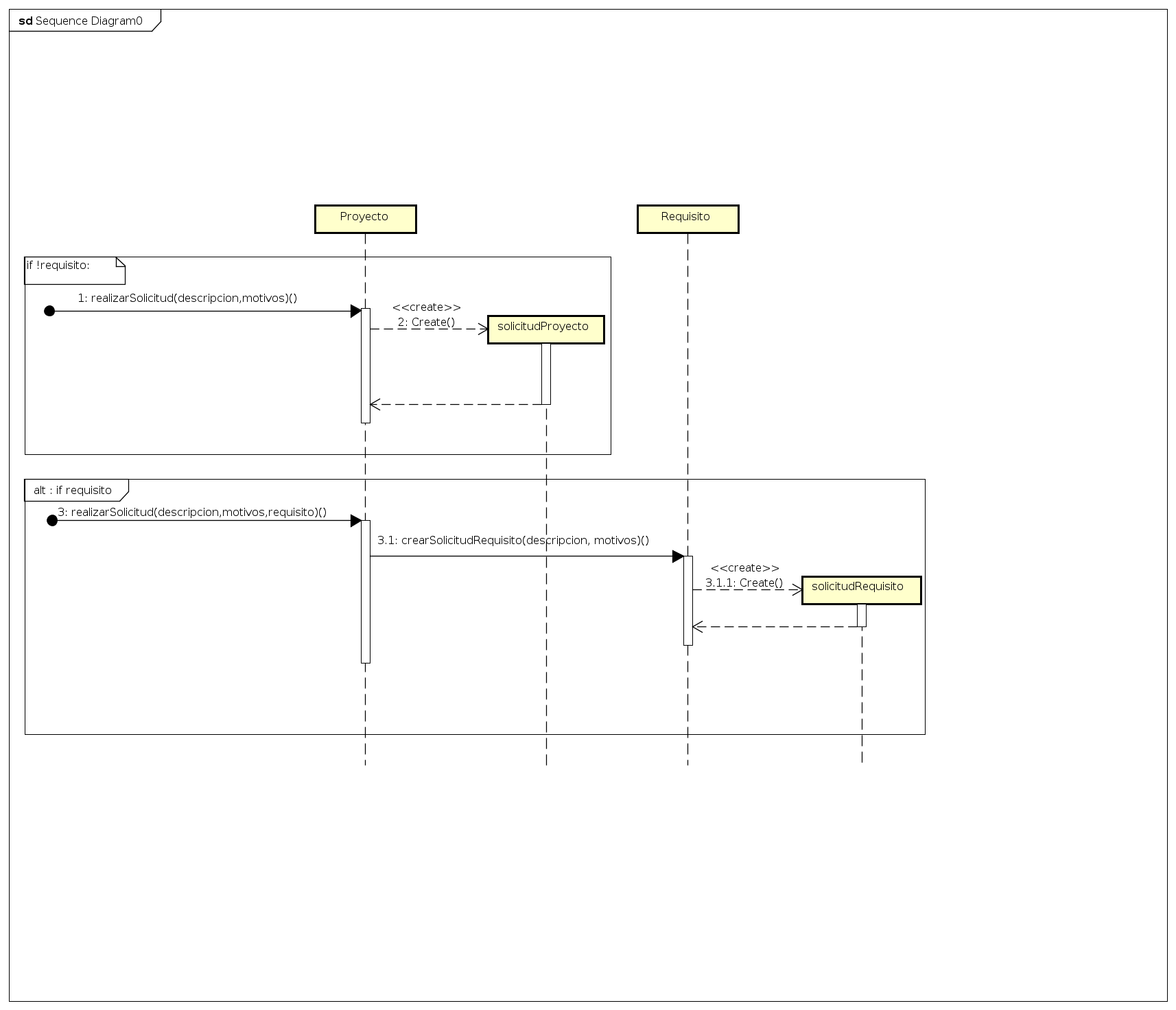
* 1. **Diagrama de Secuencia 3 – Crear Proyecto**

En la figura 03 podemos ver el proceso que se sigue en el sistema para crear un proyecto. Se invoca el método crearProyecto. Si el sistema detecta que ya existe un proyecto con ese mismo nombre, se lanzará una excepción deteniendo el proceso.

**Figura 03: Diagrama secuencia Crear Proyecto**

* 1. **Diagrama de Secuencia 5 – Crear Solicitud de Requisito**

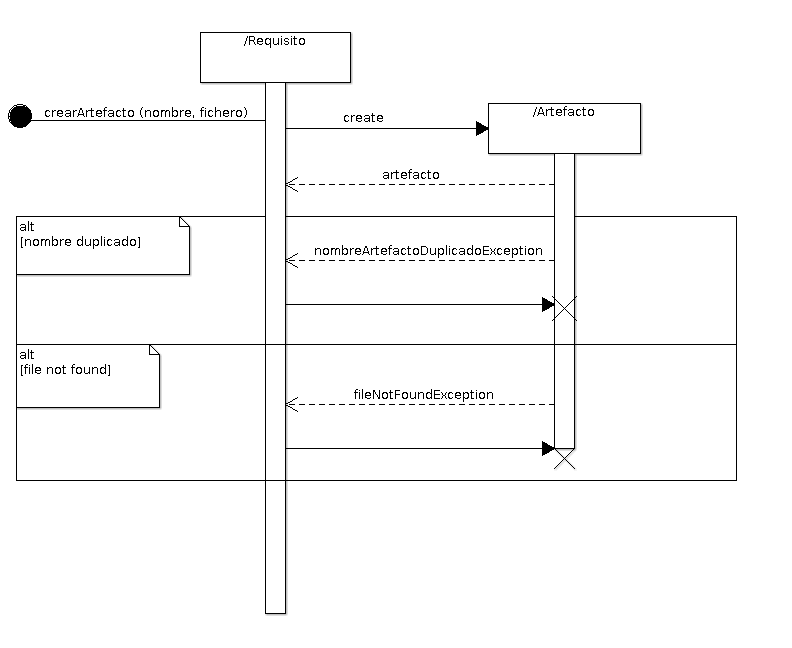
En la figura 04, se observa el diagrama para crear una solicitud. En él se muestra con claridad cómo cambia el comportamiento de la función al recibir un requisito como parámetro. En este caso, la solicitud se creará asociada al requisito en lugar de verse vinculada al proyecto completo.



**Figura 04: Diagrama secuencia Crear Solicitud de Requisito**

* 1. **Diagrama de Secuencia 7 – Crear Artefacto**

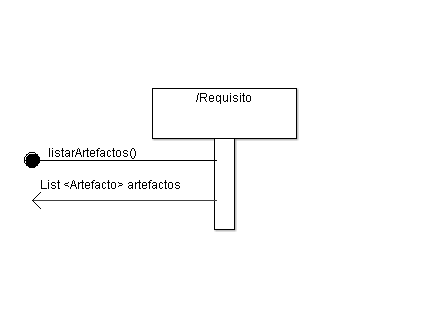
La figura 05 muestra la secuencia de métodos necesarios para crear un artefacto y la interacción entre las clases implicadas. El proceso comienza con la invocación del método crearArtefacto. Se pueden dar dos posibles casos de fallo: que ya exista un artefacto con el nombre especificado o que no se encuentre el fichero asociado al artefacto.



**Figura** **05**: **Diagrama secuencia Crear Artefacto**

* 1. **Diagrama de Secuencia 14 - Listar Artefactos**

La figura 06 es el diagrama de secuencia que representa el proceso de listado de artefactos de un requisito. El proceso es simple en comparación con los anteriores, consistiendo en una llamada a un atributo de la clase Requisito, en el cual, las instancias guardan la lista de los artefactos que tienen asociados.



**Figura 06: Diagrama Secuencia Listar Artefactos**

# 4. Glosario

|  |  |
| --- | --- |
| TÉRMINO | DESCRIPCIÓN |
| **artefacto** | Diseños, planes de prueba, resultados de pruebas, prototipos y todo aquello que se pacte con el cliente, asociado a un requisito. |
| **proyecto** | Unidad básica del sistema. Contienen requisitos y están asociados a un cliente y a un jefe de proyecto. |
| **requisito** | Unidades básicas de diseño que describen funcionalidades del sistema. Pertenecen a un proyecto y a su vez pueden contener artefactos |
| **singleton** | Patrón de diseño que permite restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. |