Contenido

[1. Objetivos 1](#_Toc167813953)

[2. Especificación y análisis de requisitos 2](#_Toc167813954)

[2.1. Actores 2](#_Toc167813955)

[2.2. Requisitos del sistema 2](#_Toc167813956)

[2.3. Requisitos funcionales 2](#_Toc167813957)

[2.4. Requisitos no funcionales 3](#_Toc167813958)

[3. Tecnologías y herramientas de desarrollo 4](#_Toc167813959)

[3.1. Tecnologías de desarrollo 4](#_Toc167813960)

[3.2. Herramientas de desarrollo 4](#_Toc167813961)

# Objetivos

Esta aplicación quiere solucionar el problema de reservar recursos (seminarios, ordenadores portátiles, aulas…) de la Universidad de Vigo en el Campus de Ourense.

El procedimiento es el siguiente: mediante una página web, los usuarios (alumnos) registrados en el sistema pueden reservar un recurso disponible para su uso en la facultad. Estos recursos son ordenadores portátiles, seminarios y aulas. Además, si algún usuario detecta un error en ese recurso, el usuario puede crear una incidencia mediante un formulario asociado a su reserva para que se solucione. Estas incidencias serán atendidas por los distintos becarios responsables de cada facultad (que también estarán registrados en el sistema) para dar soporte a los distintos recursos de cada facultad.

Para controlar que los usuarios hagan uso de los recursos se utilizarán códigos QR. Estos se generan a partir de la reserva hecha y en la conserjería de la facultad se confirmará el uso escaneándolo. En caso de realizar una reserva y no acudir, el usuario será penalizado con 2 días sin poder realizar otra.

La aplicación debe ser genérica, ya que se puede adaptar a los distintos campus de la Universidad de Vigo; además de ser usada tanto por alumnos como por becarios y administradores. Se realizará con tecnologías gratuitas y adaptadas a dispositivos móviles, ya que los usuarios se identifican con el móvil al controlar la reserva.

Una vez explicado el funcionamiento, veamos a quién utilizará la aplicación.

Tenemos tres usuarios: Alumno, Becario y Administrador.

El Alumno es el que reserva los distintos recursos y confirma esa reserva con su código QR. Además de encontrar un fallo o avería en un recurso puede crear una incidencia.

El Becario trata las incidencias reportadas que le llegan al sistema.

El Administrador es el encargado de escanear el código QR de la reserva; así como gestionar dichos recursos (crear, eliminar, modificar…).

Todos los usuarios deben estar registrados en el sistema.

Esta aplicación cumple los siguientes objetivos:

* Reservar recurso
* Crear reporte de recurso
* Solucionar reporte de recurso
* Gestionar recursos:
* Añadir un recurso
* Modificar un recurso
* Eliminar recurso
* Gestionar acceso:
* Controlar reserva del recurso

# Especificación y análisis de requisitos

Para desarrollar el software se utilizan los casos de uso, los cuales recogen las funcionalidades del sistema. En ellos se describen de manera breve las necesidades de los usuarios que interactúan con el sistema (actores).

## Actores

En el sistema se identifican tres tipos de actores:

* **Alumnos**: son los que reservan los distintos recursos disponibles para su uso. También pueden reportar fallos en estos.
* **Becarios**: atienden los fallos reportados por los alumnos.
* **Administrador**:  es el encargado de gestionar los recursos.

## Requisitos del sistema

Los requisitos del sistema se pueden definir como las características principales que puede realizar el sistema.

Identificaremos los requisitos funcionales de la forma **RFXX**, donde XX es el número que tendrá el requisito. Estos tipos de requisitos se refieren a las funciones del sistema y responden a “¿Qué hace el sistema?”.

### Requisitos funcionales

* **RF01: Reservar recurso**
* Los usuarios registrados podrán reservar recursos disponibles mediante la aplicación web.
* Se generará un código QR asociado a cada reserva realizada.
* **RF02: Crear reporte de recurso**
* Los usuarios podrán crear una incidencia sobre un recurso si encuentran fallos en él.
* **RF03: Solucionar reporte de recurso**
* Los becarios responsables de cada facultad atenderán las incidencias reportadas.
* **RF04: Gestionar recursos**
* Los administradores podrán añadir, modificar y eliminar recursos disponibles para reserva.
* **RF05: Gestionar acceso**
* Se controlará la reserva del recurso mediante el escaneo del código QR.
* Se aplicará una penalización de 2 días sin poder realizar otra reserva en caso de no acudir a la reserva realizada.

### Requisitos no funcionales

* **RNF01: Adaptabilidad y Generalidad**
* La aplicación debe ser genérica y adaptable a distintos campus de la Universidad de Vigo.
* Debe ser utilizada por alumnos, becarios y docentes.
* **RNF02: Usabilidad**
* La interfaz de usuario debe ser intuitiva y fácil de usar.
* La aplicación debe estar optimizada para dispositivos móviles.
* **RNF03: Tecnologías Gratuitas**
* El desarrollo de la aplicación se realizará utilizando tecnologías gratuitas.
* **RNF04: Registro de Usuarios**
* Todos los usuarios deben registrarse en el sistema para acceder a sus funcionalidades.
* **RNF05: Eficiencia y Rendimiento**
* La aplicación debe tener un rendimiento óptimo para gestionar reservas y recursos sin demoras significativas.
* **RNF06: Seguridad**
* Se garantizará la seguridad de los datos de los usuarios y las reservas realizadas.
* Se implementarán medidas de protección contra posibles ataques o intrusiones.
* **RNF07: Escalabilidad**
* La aplicación debe ser escalable para manejar un aumento potencial en el número de usuarios y recursos gestionados.

# Tecnologías y herramientas de desarrollo

## Tecnologías de desarrollo

* **CSS (Cascade Style Sheet)**: usadas para la apariencia y estilo de la aplicación Web. Su finalidad es crear unos estilos de diseño. CCS está altamente compenetrado con HTML para decorar las vistas del sistema.
* **HTML (Hyper Text Markup Language)**: lenguaje de marcado predominante para la creación de páginas web estáticas. Basado en el empleo de etiquetas. Se ha optado por su uso por su facilidad de manejo, curva de aprendizaje relativamente baja y el gran soporte online debido a su gran uso.
* **JavaScript**: lenguaje de programación basado en prototipos, orientado a objetos e imperativo con múltiples usos (scripting, interfaces web, servidores...). Usado para validar los datos introducidos por los usuarios.
* **PHP (Hypertext Preprocessor)**: para crear la base y gran parte de la funcionalidad del sistema se ha optado por PHP, uno de los lenguajes de desarrollo web más conocidos y usados, además de tener un gran soporte.
* **SQL (Structured Query Language)**: lenguaje de manejo y gestión de base de datos por antonomasia, de ahí que se haya optado por usarlo para gestionar la base de datos del sistema.

## Herramientas de desarrollo

* **Chrome y Firefox**: son los navegadores web utilizados para probar la web. Han sido elegidos por su amplia utilización y su rapidez.
* **GitHub**: es un sistema de control de versiones gratuito y de código abierto. Se ha elegido por ser el más popular en su ámbito. Se ha empleado para realizar el control de las versiones de forma ordenada.
* **Microsoft Word**: Este es el software de procesamiento de texto para redactar este documento. Ha sido elegido por mi familiaridad y su calidad.
* **Visual Studio Code**: editor de texto. Utilizado para la implementación del código, tanto de PHP, HTML y CSS. Se da la casualidad de que ninguno de estos 3 lenguajes es “compilable”, por lo que con un editor de texto podemos abarcar la implementación del sistema sin necesidad de ningún IDE.
* **Visual Paradigm**: herramienta de diseño de software. Ha sido utilizado para la creación de los diagramas del sistema y otros artefactos a lo largo de todo el proceso. Se ha elegido por mi familiaridad.