

# **Proyecto Web: Soluciones Vecinales**

Francisco Javier Sueza Rodríguez

20 de febrero de 2025

**Centro:** IES Aguadulce  
**Ciclo Formativo:** Desarrollo Aplicaciones Web (Distancia)

# Índice

<b>1. Introducción</b>	<b>3</b>
1.1. Presentación . . . . .	3
1.2. Contexto . . . . .	3
1.3. Planteamiento del Problema . . . . .	3
1.4. Visión General . . . . .	4
<b>2. Análisis de Requisitos</b>	<b>4</b>
2.1. Introducción . . . . .	4
2.1.1. Propósito . . . . .	4
2.1.2. Alcance . . . . .	5
2.1.3. Definiciones, Siglas y Abreviaturas . . . . .	5
2.1.4. Referencias . . . . .	5
2.1.5. Visión General . . . . .	5
2.2. Descripción General . . . . .	6
2.2.1. Perspectiva del Producto . . . . .	6
2.2.1.1. Interfaz de Usuario . . . . .	6
2.2.1.2. Interfaces con Hardware . . . . .	6
2.2.1.3. Interfaces con Software . . . . .	6
2.2.1.4. Interfaces de Comunicación . . . . .	7
2.2.1.5. Restricciones de Memoria . . . . .	8
2.2.1.6. Requerimientos de Adecuación al Entorno . . . . .	8
2.2.2. Funciones del Producto . . . . .	8
2.2.3. Características de los Usuarios . . . . .	10
2.2.4. Restricciones de Diseño . . . . .	10
2.2.4.1. Lenguajes de Programación . . . . .	10
2.2.4.2. Software para el Desarrollo . . . . .	10
2.2.4.3. Control de Calidad y Testeo . . . . .	11
2.2.4.4. Seguridad de la Aplicación . . . . .	11
<b>Anexo A. Descripción de la Interfaz de Usuario</b>	<b>12</b>
<b>Glosario</b>	<b>16</b>
<b>Referencias</b>	<b>17</b>

# 1. Introducción

## 1.1. Presentación

En este documento se detalla el proyecto final desarrollado para el **CFGS Desarrollo de Aplicaciones Web** en el **IES Aguadulce**. El proyecto elegido ha sido una aplicación web para la **gestión y administración de comunidades de vecinos**, llamada **SolucionesVecinales**.

El objetivo de esta aplicación es facilitar la **gestión de comunidades de vecinos** por parte tanto de los administradores de la propiedad como de los inquilinos o propietarios, ofreciéndose de **forma gratuita** y poniendo especial énfasis en las **accesibilidad** y **facilidad de uso**, con una interfaz web amigable y accesible desde cualquier dispositivo que disponga una navegador web.

La aplicación trabaja con **3 tipos diferentes de usuarios**, incluyendo administradores, usuarios y visitantes, siendo el público objetivo los administradores de propiedad y presidentes de comunidad así como todos los inquilinos y propietarios de las comunidades de vecinos. Los diferentes usuarios tendrán diferentes permisos y responsabilidades.

A lo largo de este documento, desgranaremos todas los requisitos de la aplicación, el diseño elegido y todos los detalles del proceso de desarrollo y testeo.

## 1.2. Contexto

Esta aplicación es un **proyecto personal** que se ha decidido realizar tras el análisis de diferentes aplicaciones similares y ver determinadas carencias que pueden dificultar la inclusión de todos los potenciales usuarios. Hay que tener en cuenta, que los usuarios objetivos son un grupo muy heterogéneo donde podemos encontrar personas con diferente formación, conocimientos sobre tecnologías o con problemas de accesibilidad, así como gente con dificultades económicas.

Por este motivo, se ha decidido realizar una **aplicación gratuita**, centrándonos en la **accesibilidad** y **facilidad de uso**, que ayude a cualquier potencial usuario a usar la aplicación e implicarse en la gestión de comunidad de vecinos.

## 1.3. Planteamiento del Problema

La gestión de comunidades de vecinos es una tarea compleja que requiere realizar un conjunto de gestiones administrativas asegurando la participación de los vecinos y de una forma transparente. En este aspecto, nuestra aplicación debería permitir que se realicen las siguientes tareas:

- **Gestión de documentos:** gestión de los documentos generados en los diferentes procesos, como resolución de incidencias, documentos sobre juntas vecinales, etc. Permitiendo que estos puedan ser accedidos por todos los vecinos de la comunidad.
- **Reserva de espacios comunes:** automatizar la reserva y gestión de espacios comunes permitiendo a los vecinos reservar o cancelar reservas de dichos espacios, así como acceder a una lista con los espacios y el estado actual de reserva.
- **Gestión de Incidencias:** se debe permitir la creación de incidencias por parte de los vecinos así como su administración por parte del presidente de la comunidad o el administrador de fincas.
- **Gestión de Usuarios y Comunidades:** la aplicación debe permitir la creación, baja, actualización y consulta de usuarios con diferentes roles así como de comunidades de vecinos.

- **Gestión de Finanzas:** la aplicación debe realizar, de forma automática, cálculos para administrar económicamente la comunidad, calculando el balance anual a partir de las cuotas que se están pagando y las diferentes facturas que se han pagado.

Además de estas tareas, hay que garantizar la seguridad de la información almacenada, ya que es información sensible, así como el proceso para realizar estas tareas sea sencillo e intuitivo, además de accesible.

## 1.4. Visión General

Este documento se compone de 6 secciones principales que tratan diferentes aspectos del proyecto. Para facilitar al interesado la navegación por éste, vamos a explicar brevemente en qué consiste cada sección del documento:

- **Análisis de Requisitos:** en esta sección se especifican los requisitos, tanto funcionales como no funcionales de la aplicación, especificando qué es lo que se quiere conseguir con este proyecto y cuáles son las restricciones que vienen impuestas, bien por el problema a tratar o por los propios requisitos.
- **Hardware y Software Necesario:** en esta sección se especifica el hardware necesario para realizar el desarrollo y poner en funcionamiento la aplicación así como el software que se va a necesitar para este mismo propósito. También se especifica el presupuesto y los costes de hosting para la aplicación.
- **Casos de Uso:** en esta sección se realiza un análisis de los principales casos de uso, empleando diferentes diagramas, así como una descripción más detallada de los casos de uso generales.
- **Diseño de la Interfaz:** en esta sección se especifica el diseño de la interfaz de usuario, mostrando todas las páginas de la aplicación así como la relación entre estas y sus diferentes elementos.
- **Diseño de la Base de Datos:** en este apartado se especifica el diseño de la base de datos, empleando un esquema Entidad-Relación así como el paso a tablas de esta, explicando los puntos que sean oportunos sobre las decisiones de diseño tomadas. Además, se incluyen dos scripts, uno para la creación de la base de datos y otro para poblarla con datos.
- **Diagrama de Componentes:** en esta sección se muestra el diagrama de componentes de la aplicación, poniendo de relieve la arquitectura de la aplicación así como las interacciones entre los diferentes elementos y sistemas.

## 2. Análisis de Requisitos

### 2.1. Introducción

En esta sección se va a definir la **especificación de requerimientos** que establece los requisitos funcionales y no funcionales de la aplicación SolucionesVecinales, realizando un análisis del funcionamiento esperado de la aplicación y estableciendo unos límites para el diseño de ésta.

#### 2.1.1. Propósito

El propósito de esta sección es **establecer los requisitos**, tanto funcionales como no funcionales, de la aplicación que se va a desarrollar, realizando un **análisis exhaustivo de su funcionamiento** y estableciendo las restricciones necesarias para que la aplicación cumpla su funcionalidad de forma correcta,

segura y eficiente. Estos requisitos nos ayudarán a planificar el proceso de desarrollo así como a tomar las decisiones oportunas durante el proceso de diseño de la aplicación.

Esta sección va dirigida principalmente al **equipo de desarrollo**, que serán los encargados de elaborar el diseño de la aplicación a partir de los requerimientos y restricciones establecidos en este documento, aunque teniendo en cuenta que también esta incluido en el documento del proyecto en general cabe la posibilidad de que tengan acceso a él otros interesados, como por ejemplo, posibles inversores.

### 2.1.2. Alcance

Nuestro producto se llama **SolucionesVecinales** y es una aplicación web para la **gestión de comunidades de vecinos**. El fin de nuestra aplicación es ayudar a todos los integrantes en una comunidad de vecinos a realizar diferentes gestiones relacionadas con las administración de su comunidad, de forma simple y sencilla.

Es una aplicación que se **ofrece de forma gratuita**, cuya meta principal es llegar al mayor número de usuarios, implementando buenas **políticas de accesibilidad** y ofreciendo una **interfaz sencilla e intuitiva** que pueda usar cualquiera persona, incluso aquellos que no están familiarizados con las nuevas tecnologías.

Partiendo de esta base, la aplicación deberá **ayudar a realizar las tareas** más comunes en las que deben participar los vecinos en una comunidad, como el acceso de documentos, creación de incidencias, consulta de las cuentas de la comunidad, reserva de espacios comunes, etc.

### 2.1.3. Definiciones, Siglas y Abreviaturas

Las definiciones, siglas y abreviaturas se incluyen al final del documento, dentro de un Glosario, donde se incluyen todos los términos tanto de esta sección como del resto de secciones.

### 2.1.4. Referencias

La referencias, al igual que el glosario, tanto de esta sección como del resto de secciones se incluyen al final de documento, para no duplicar secciones. Además, en cada texto que haga referencia a alguna página o documento, se incluirá la cita adecuada apunta a dicha referencia, para facilitar su acceso.

### 2.1.5. Visión General

Este documento se divide principalmente en 4 secciones, incluyendo la actual. Cada sección describe un aspecto diferentes de los requisitos del software y son las siguientes:

- **Introducción** (Sección 2.1): en esta primera sección se realiza una introducción tanto al software como a este documento, explicando su propósito, alcance, etc...
- **Descripción General** (Sección 2.2): en esta sección se realiza una descripción del software que se va a desarrollar, incluyendo su funcionalidad, restricciones, características de los usuarios potenciales, interfaces, etc. Nos sirve como base para especificar los requisitos en el siguiente apartado.
- **Requerimientos Específicos** (Sección 2.3): en la tercera sección vamos a establecer los requisitos específicos del software, extraídos de la descripción general y separándolos en requisitos funcionales y no funcionales.

- **Requerimientos de Documentación** (Sección 2.4): en esta sección de la parte de requisitos se especifican los requisitos de documentación que va a tener el software, incluyendo manuales, ayuda en línea, instalación, etc.

## 2.2. Descripción General

En esta sección se van a describir todos los factores y restricciones que afectan al producto, estableciendo una base para la definición de los requisitos específicos en la siguiente sección, incluyendo una perspectiva general del producto, así como las restricciones que se deben aplicar en su diseño.

### 2.2.1. Perspectiva del Producto

En primer lugar vamos a realizar una descripción del producto, especialmente de las diferentes interfaces que se van a emplear y su interacción con los diferentes elementos y sistemas con los que deberá interactuar nuestro software.

#### 2.2.1.1. Interfaz de Usuario

Debido a que las especificaciones sobre al interfaz de usuario son más extensas, se han incluido como un anexo. Estas especificaciones se pueden consultar, en concreto, en el [Anexo A](#).

#### 2.2.1.2. Interfaces con Hardware

Nuestro software **no interactúa directamente con el hardware** de los dispositivos, ya que en la parte del cliente la interacción se realiza con el navegador web, mientras que en la parte del servidor la interacción se realiza con el servidor web. Estos, a su vez, realizarán las interacciones necesarias con el SO.

Sí cabe destacar en esta sección, ya que no veo otra donde pueda incluir esta información, que **nuestro software** se deberá usar desde **cualquier dispositivo** que cuente con un **navegador web**, por lo que deberá realizarse el diseño teniendo esto en cuenta. Realmente esto solo afecta a la estética de la aplicación en cada dispositivo, y no en su funcionamiento, y tampoco requiere del empleo de ninguna interfaz específica, por lo que solo afectará a la **interfaz de usuario**.

#### 2.2.1.3. Interfaces con Software

Nuestra aplicación **no usa ninguna interfaz externa** para comunicarse con ningún software externo.

La **única interacción** que cabe mencionar es la que realiza la parte de cliente de nuestra aplicación con el **navegador web** del usuario, aunque en este aspecto no se emplea ninguna interfaz específica para llevar a cabo esta interacción, si no que se realiza a través de los lenguajes empleados para desarrollar el propio cliente, como HTML, CSS, Javascript, etc.

En el **lado del servidor**, nuestra aplicación se ejecutará en un servidor web, aunque como hemos comentado, no se va a emplear ninguna interfaz para comunicarse con el. Para ejecutar nuestra aplicación vale cualquier servidor web, aunque si hay que tener en cuenta un detalle, y es que cuando la aplicación se este **ejecutando en producción**, el servidor deberá tener soporte para **SSL**, ya que nuestra aplicación va a emplear el protocolo **HTTPS** para comunicarse.

#### 2.2.1.4. Interfaces de Comunicación

Para que nuestra aplicación funcione correctamente **necesita comunicarse mediante una red**, ya sea local o internet, realizando la comunicación entre nuestro cliente, con el navegador del usuario, y nuestro servidor. Para llevar a cabo esta comunicación, se va a emplear el protocolo **HTTP**. Especificaciones El **protocolo HTTP** es el estándar para la comunicación web desde hace muchos años, inventado por Tim Berners-Lee en 1989, es un protocolo de la capa de aplicación que es la base de la comunicación en la World Wide Web. [1]

En este documento **no vamos a discutir** ni exponer las **especificaciones** de todo el **protocolo HTTP**, no siendo relevante para nuestra aplicación ya que el uso de este protocolo se va a realizar mediante funciones predefinidas por los lenguajes sin tener que preocuparnos de cómo se implementa dicho protocolo, aunque sí vamos a comentar un par de puntos que sí tiene que ver con nuestra aplicación.

En primer lugar, hay que hablar de los **verbos HTTP**, que son un conjunto de métodos que se usan para solicitar, desde el cliente, diferentes tipos de acciones que debe realizar el servidor, como puede ser mandar información, modificarla, eliminarla, etc... Los verbos HTTP que podemos emplear son GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE y PATCH. [2]

Cada método tiene una funcionalidad diferente, con diferentes ventajas e inconvenientes, pero nosotros no vamos a describir aquí todos sino que nos vamos a centrar en los únicos que nos interesa y que vamos a usar en nuestra aplicación, que son **GET, POST, PUT y DELETE**. Las principales características de estos métodos son las siguientes:

- **GET**: este método se emplea para realizar **peticiones al servidor** sobre el **estado representacional de algún recurso**. Las peticiones con este método se pueden almacenar en una caché y la información de la petición se pueden ver en la URL que se le pasa al servidor. Además es un método seguro, ya que no altera información del servidor e **idempotente**. [3]
- **POST**: este método se emplea para **enviar información al servidor**, la cual se envía en el **cuerpo de la petición**. El resultado puede ser alguna acción realizada por el servidor como comprobar que los datos son correctos, crear nuevos datos o modificarlos. Es un método no seguro, ya que modifica información en el servidor, y tampoco es cacheable ni idempotente. [4]
- **DELETE**: con éste método se le indica al servidor que elimine un recurso específico. Este método no es seguro ni cacheable, pero sí es idempotente. [5]
- **PUT**: este método crea o modifica un recurso en el servidor añadiendo el contenido en el cuerpo de la solicitud. La principal diferencia entre PUT y POST es que PUT es idempotente. [6]

Estos métodos tienen que tenerse claros en el proceso de desarrollo, ya que la **API** de nuestra aplicación deberá manejar correctamente las diferentes peticiones que se hagan con estos métodos, y nuestro cliente deberá realizar las peticiones con el método adecuado. Se especificará más detalladamente en el apartado de restricciones.

Cabe destacar que nuestra aplicación usará **HTTPS**, aunque esto no va a afectar en cómo van a tratarse los métodos ni a procesarse, sino en cómo se realizará la conexión, que tiene más que ver con la configuración del servidor.

Por otro lado, debemos hablar de los **códigos HTTP de respuesta**. Estos códigos son los que deberá responder el servidor a las diferentes peticiones que realice el cliente. No vamos a explicar cuáles son todos los códigos, ya que sería muy extenso, simplemente comentar que en todo momento se deberá enviar el código adecuado al resultado de la operación solicitada por el cliente y con mensajes descriptivos que expliquen bien el motivo de ese error.

Para consultar con más detalle los diferentes códigos que podemos encontrar podemos visitar la [web de MDN](#) sobre dichos códigos, donde hay una lista extensiva con todos y cada uno de los códigos empleados.

#### 2.2.1.5. Restricciones de Memoria

La **restricciones de memoria** en nuestra aplicación son muy dispares, ya que las que se aplican al servidor no son las mismas que las que se aplican al cliente, por lo que vamos a tratarlas las dos de forma individual escuetamente.

En nuestro **servidor** las restricciones tanto en memoria primaria como secundaria son mínimas. La cantidad que se use de ambas va a estar estrechamente **relacionada con el número de usuarios** que tenga nuestra aplicación, ya que el principal uso de memoria principal va a ser ocupado por las peticiones de los usuarios y las acciones que tenga que llevar, en consecuencia, el servidor sobre el sistema. Por otro lado, nuestra aplicación debe almacenar una gran cantidad de datos por cada comunidad, incluyendo archivos multimedia como es el caso de imágenes, por lo que la cantidad almacenamiento secundario que necesitaremos también va a depender en gran medida del número de usuarios que tengamos.

Por otro lado, nuestro **cliente** si va a tener **fuertes restricciones**, especialmente en el uso de **memoria principal**, ya que no se espera uso de memoria secundaria más allá de lo que puedan ocupar las cookies y otros elementos de este tipo. Pero en la memoria principal si tendremos restricciones, ya que hay que tener en cuenta que nuestra aplicación debe ejecutarse en muchos dispositivos que tienen recursos muy limitados, como pueden ser los dispositivos móviles, especialmente los más antiguos. En este aspecto **deberá minimizarse** el uso de **memoria principal**, evitando efectos indeseados como excesivos renderizados o utilizando imágenes comprimidas.

#### 2.2.1.6. Requerimientos de Adecuación al Entorno

La aplicación **no tiene requerimientos especiales** para su ejecución en diferentes entornos, ya que esta se va a ejecutar en cualquier servidor web que tenga soporte para SSL. Respecto a la **visualización en diferentes tipos de dispositivos**, en la **sección de restricciones de diseño** se añadirán detalles sobre como afrontar esta situación.

#### 2.2.2. Funciones del Producto

A lo largo de este documento ya hemos comentado que nuestro software, **SolucionesVecinales**, es una aplicación de **gestión de comunidades de vecinos**, pero no hemos especificado cual es su funcionamiento, así que en esta sección vamos a realizar una descripción general del funcionamiento de la aplicación.

- La aplicación deberá **gestionar los diferentes usuarios** permitiendo que éstos se registren en el sistema como usuarios o administradores y que accedan a las diferentes herramientas para gestionar su comunidad, así como a la modificación de su perfil o su eliminación.
- Los **administradores** podrán añadir nuevas comunidades, donde los usuarios podrán inscribirse. El **sistema generará automáticamente** una **solicitud de inscripción** cuando se registre un usuario a la comunidad con la dirección del usuario. El **administrador deberá aprobar** la solicitud para que el usuario quede inscrito.



- Una vez que los **usuarios o administradores** estén inscritos en una comunidad, tendrán acceso al sistema de gestión de comunidades, donde podrán realizar diferentes acciones en su comunidad, además de tener la opción de editar su perfil o eliminar su usuario desde su página de perfil.
- Dentro de las **herramientas** que provee nuestro software para **gestionar la comunidad**, tenemos las siguientes:
  - **Inicio:** en la pantalla de inicio de gestión de comunidad se mostrará información de la comunidad, así como información relativa a los elementos creados por el usuario, como pueden ser incidencias, documentos descargados, reservas de espacios comunes, etc.
  - **Tablón de Anuncios:** en el tablón de anuncios el administrador podrá **crear mensajes** que van dirigidos a los miembros de la comunidad. Los mensajes se mostrarán como una **lista** con la fecha de creación debajo y todos los usuarios podrán consultarlos desde su tablón de anuncios. Solo el **administrador** podrá **crear** o **eliminar** mensajes.
  - **Documentos:** en esta sección el **administrador** subirá los diferentes **documentos** relativos a la **gestión de su comunidad**, como pueden ser resoluciones judiciales, legislación, actas de reuniones. El resto de usuarios podrán visualizar y descargar dichos archivos. Los archivos se **mostrarán como un conjunto de iconos** con el nombre debajo y el tamaño. Solo el **administrador** puede subir o eliminar archivos de esta sección.
  - **Incidencias:** esta sección permitirá a todos los usuarios la **creación de incidencias**, para ponerlas en conocimiento de los administradores y que tomen las medidas oportunas para su resolución. Cuando un usuario ingrese en esta página podrá **ver** todas las **incidencias creadas por el mismo** o por otros miembros de la comunidad que hayan **marcado la incidencia como pública**.  
  
Las incidencias pueden estar en tres estados: “Creada”, “En Proceso” o “Cerrada”, según el momento de procesamiento en el que se encuentre ésta. Las incidencias se mostrarán en una tabla, con su nombre, descripción, fecha de creación y estado. El **administrador** podrá **cambiar el estado** de las incidencias, además de eliminarlas.
  - **Espacios Comunes:** en esta sección los usuarios podrán **reservar los espacios comunes** que sea objeto de reserva en la comunidad. Los espacios comunes serán **añadidos por el administrador**, que será también el que establezca el horario en el que se pueden reservar.  
  
Los **usuarios**, podrán consultar el estado de reserva de un espacio común y realizar una reserva en el horario y día que este disponible. Cuando un usuario tenga reservas realizadas, aparecerán debajo de los espacios comunes, permitiéndole borrar o cambiarlas.
  - **Finanzas:** la sección de finanzas **mostrará una tabla con todos los ingresos y gastos** previstos para este año, aunque también se podrán consultar los del ejercicio anterior. El administrador podrá añadir o eliminar registros, lo que provocará que la aplicación haga el cálculo de nuevo, mostrando el nuevo resultado.

Estás son las principales funcionalidades que tiene nuestro software. **No es algo inamovible**, y en un futuro podría (y debería) plantearse la adición de nuevas funcionalidades. Para ello será interesante crear una canal de feedback con los usuarios y escuchar que tienen que decir, especialmente, que echan en falta.

### 2.2.3. Características de los Usuarios

Los usuarios objetivos de nuestra aplicación son un **grupo muy heterogéneo**, que abarca a **cualquier persona** que viva en una comunidad de vecinos, es decir, la gran mayoría de la población. Por ello vamos a encontrarnos usuarios con todos los niveles de experiencia, nivel educacional, edad, etc.

Teniendo esto en cuenta, la aplicación debe ser lo suficientemente simple e intuitiva en su uso para que los usuarios con menos experiencia, nivel educativo o los que tienen problemas de accesibilidad puedan usarla correctamente.

### 2.2.4. Restricciones de Diseño

En este apartado vamos a realizar una lista con todas las restricciones de diseño que deben aplicarse a la hora de desarrollar la aplicación, desde lenguajes de programación y herramientas de desarrollo, pasando por políticas de testeo, seguridad de datos, etc...

#### 2.2.4.1. Lenguajes de Programación

El lenguaje de programación principal empleado, tanto en el **frontend** (cliente) como en el **backend**, será **Typecript**, ya que proporciona toda la funcionalidad de Javascript con elementos interesantes como la definición de tipos o interfaces. En el **backend** se usará, además, **SQL** para trabajar con la base de datos.

#### 2.2.4.2. Software para el Desarrollo

Para el desarrollo de la aplicación se usarán diferentes **librerías**, **entornos de ejecución**, etc. En nuestro caso, se va a especificar el software que es de obligatorio uso para el desarrollo de nuestra aplicación.

- **FrontEnd**: en esta parte se va a emplear **React.js** para desarrollar el cliente empleando componentes. Además se debe emplear el preprocesador de CSS **SASS**, para crear archivos CSS modulares y reutilizables.

Después de un análisis más profundo de la aplicación **se ha descartado el uso de Redux**, ya que con **Context** de React se podrá realizar la misma funcionalidad disminuyendo la complejidad.

- **BackEnd**: en la parte de backend se va a emplear **Node**, para poder ejecutar Typescript en el lado del servidor junto con **Express**, que nos facilitará la creación de la API. Además, se empleará para la base de datos **Postgresql**, junto con **Prisma**, un **ORM** que facilitará el trabajo con la base de datos.
- **Herramientas de Desarrollo**: cada programador podrá usar el editor o IDE que sea de su preferencia, pero si se deberá usar **Prettier**, para establecer un estándar en el estilo código y **Eslint**, para detectar errores y adherirse a buenas prácticas.
- **Gestión del Proyecto**: para la gestión del proyecto se van a emplear **Github**, como repositorio de software y **Trello**, que se empleará para la gestión de las tareas que hay que realizar a modo de tablero Kanban.

No se va a especificar el software para el testing ni el control de calidad, ya que este se especificará en su respectiva restricción.

#### 2.2.4.3. Control de Calidad y Testeo

Se tiene que implementar una **política de control de calidad** que incluya el testeo continuo de la aplicación a varios niveles, para asegurar su calidad y localizar errores de forma más rápida. Esta política consiste en los siguientes puntos:

1. El desarrollo de cada módulo de software se realizará empleando la metodología **TDD, implementando primeramente los test** con la funcionalidad básica y desarrollando posteriormente el código que cumpla con dicha funcionalidad. Posteriormente se **refactorizará** teniendo en cuenta otros aspectos como la performance, la eficiencia, seguridad, etc..
2. Para realizar los test unitarios se empleará **react-testing-library** en la parte del cliente y **Jest + Superset** para la parte del servidor.
3. Se realizarán un proceso de **integración continua** empleando para ello **Github Actions**, siendo prioridad que un trozo de software que se ha desarrollado pase todos los test y haya una **cobertura del 100 %** del código con los test creados.
4. Se integrará **Cypress** en este proceso integración continua para la realización de los e2e.
5. El código de la aplicación estará incluida en **Code Climate**, donde se deberá revisar periódicamente y realizar los cambios indicados por este software para tener un código de calidad. Esto será complementario, como se ha comentado en otro punto, con el uso de **linters** como **Eslint**.
6. Se seguirán las **buenas prácticas** tanto en el desarrollo del cliente como del servidor, siguiendo los estándares y recomendaciones de la **W3C**.

#### 2.2.4.4. Seguridad de la Aplicación

A la hora de tratar con los datos de los usuarios y las comunidades hay que ser extra precavido, y tener en cuenta unas restricciones para garantizar la seguridad de dichos datos. Estas restricciones son:

1. Los **datos que se obtenga de formularios** deben ser **doblemente validados**. Por un lado, se procesarán en el cliente, comprobando que tienen el formato correcto y eliminando aquellos que tengan ciertas palabras que pueden considerarse maliciosas, especialmente para ataques de **inyección SQL**. En una segunda parte se **volverán a validar en el servidor** antes de incluirlos en la base de datos.
2. **Nunca** se **almacenarán las contraseñas en texto plano** en la base de datos. Para su almacenamiento, se empleará alguna función de **hasing** como **bcrypt**, almacenando el resultado en vez de la contraseña.
3. Las **contraseña de los usuarios** deberán tener un **longitud de 12 caracteres o más**, e incluir algún **número y signo de puntuación**.
4. Nuestra base de datos, que estará físicamente en el mismo servidor que el servidor web, **solo debe aceptar consultar locales**, para evitar determinados tipos de ataques.
5. Las **peticiones por segundo** al servidor **estarán limitadas**, para evitar ataques de tipo **DOS**.
6. El **software** que se emplee en el desarrollo y necesario para la ejecución de la aplicación deben estar siempre **actualizado a la última última versión**.
7. Solo se permitirán conexiones **HTTPS** con la aplicación.

## Anexo A Descripción de la Interfaz de Usuario

En este anexo se va a describir la interfaz de la aplicación, especialmente a un nivel gráfico, ya que a nivel de funcionamiento se va a describir con más detalle en otro punto de esta sección.

SolucionesVecinales es una aplicación web, por lo que tendrá una **interfaz web** que podrá ser accedida desde cualquier navegador y prácticamente desde cualquier dispositivo. La interfaz estará compuesta por un conjunto de páginas web que ayudarán a los usuarios a utilizar las diferentes herramientas que ofrece la aplicación para gestionar la comunidad de vecinos.

Las **páginas** de las que estará compuesta nuestra interfaz son las siguientes:

- **Página Principal:** esta es la primera interfaz con la que se encuentra un usuario al ingresar en nuestra aplicación web. La función principal de esta pantalla es la de **ofrecer información sobre la aplicación**, así como proporcionar un **menú con diferentes opciones** que, además de permitir la navegación por las diferentes secciones permita a un usuario **registrarse** o **realizar login**.

Las secciones de las que estará compuesta esta interfaz serán las siguientes:

- **Cabecera:** aquí se mostrará una imagen con un eslogan, el nombre de la aplicación y un elemento **CTA** para que los usuarios se registren o realicen login.
- **Barra de Menú:** esta barra, que se sitúa inicialmente encima de la cabecera y que debe permanecer en todo momento en la parte superior de la pantalla, esta compuesta del **logo de la aplicación** en el lado izquierdo y de un **menú** en el lado derecho. El menú cambiar las opciones que ofrece dependiendo de si el usuario esta registrado o no.
- **Secciones:** esta página contendrá, como mínimo, **3 secciones**, donde se explicará un poco el **propósito** del software, sus principales **características**, lo que nos diferencia de la competencia, etc. El número de secciones puede variar, pudiendo ser necesario agregar más. Las secciones deben ser **visualmente atractivas**, incluyendo imágenes e iconos y con una disposición adecuada.

Una sección que deberá ser **obligatoria**, es la sección “**Contacta con nosotros**”, compuesta por un formulario y que permitirá a cualquier usuario, registrado o no, contactar con el equipo de desarrollo.

- **Pie de Página:** en el pie de página se añadirá una lista de enlaces con las diferentes secciones de la página, un mapa del sitio web y un conjunto de enlaces de redes sociales de la aplicación o en su defecto del equipo de desarrollo.

Tanto la **barra de menú** superior como el **pie de página** se mantendrán **visibles en todas las páginas** de la aplicación, cambiando en ciertos aspectos u ofreciendo diferentes opciones.

- **Página de Login:** a esta pantalla se accede desde la pantalla principal cuando el usuario pulsa en el enlace de login/registro del menú de la barra principal y su propósito es permitir que los usuarios ingresen al sistema. La pantalla inicial esta compuesta por un **formulario** de login, que debe aceptar un **email** y una **contraseña** como entrada, con el botón enviar en la parte inferior. Además, incluirá un **enlace** debajo del botón, centrado, que permitirá al usuario **registrarse**, en caso de que no este registrado, desplegando un nuevo formulario que veremos en detalle a continuación.
- **Página de Registro:** en esta pantalla, que se accede desde la pantalla de login, se muestra un formulario que **permitirá a un usuario registrarse** en el sistema. Este formulario estará compuesto por diferentes campos que permitirán al usuario introducir la información necesaria para registrarse, como puede ser Nombre, Correo, Dirección, rol, etc. Debajo, aparecerá el **botón registrarse**

que permitirá al usuario registrar todos sus datos.

- **Página Principal de Comunidad:** una vez que el usuario haya ingresado al sistema, se mostrará la pantalla de comunidad. El propósito de esta pantalla es servir como **puerta de entrada a la gestión de la comunidad**. Esta interfaz estará compuesta por los siguientes elementos:

- **Barra de Menú:** similar a la que vemos en la página principal, pero las opciones de menú “Login/Registrarse” habrá cambiado por el **icono de un usuario**. Este icono mostrará un **menú vertical** con las opciones “**Perfil**” y “**Logout**”, que permitirán acceder a la información de perfil y hacer **logout** respectivamente.
- **Menú Lateral:** un menú lateral, situado a la izquierda, que mostrará las opciones relativas a la comunidad que este cargada. Las opciones variarán según si el usuario tiene el rol de administrador o de usuario, pero las opciones comunes a ambos son las siguientes:
  - **Inicio**
  - **Tablón de Anuncios**
  - **Documentos**
  - **Incidencias**
  - **Espacios Comunes**
  - **Finanzas**

Adicionalmente, si el usuario tiene el **rol administrador**, se mostrará la opción “**Administrar**”, que permitirá al administrador gestionar los aspectos principales de la comunidad desde una misma pantalla. La cual se describirá más adelante.

El menú lateral se mostrará solo en caso de que el usuario, independientemente de su rol, este inscrito en una comunidad. En caso contrario no se mostrará.

- **Zona de Contenido:** en esta zona, que estará ubicada en el centro, se mostrará la información que se quiere mostrar al usuario. Cuando un **usuario inscrito en una comunidad** acceda a la página principal de comunidad, aquí se mostrará la **información** de ésta, como nombre, dirección y una imagen. Además, se mostrarán las **incidencias sin resolver** creadas por el usuario y las **reservas pendientes** que ha realizado, debajo de la información anterior, en formato de lista.

En esta zona de la interfaz se mostrará toda la información que el usuario solicite y es donde se permitirá interactuar con la comunidad. Seleccionando alguna de las opciones del menú, los **principales elementos** que se mostrarán en esta zona y que permiten **realizar las acciones sobre la comunidad** son las siguientes:

- **Inicio:** cuando un usuario pinche en esta opción cargará la información por defecto de la página principal de comunidad, que se ha descrito en el párrafo anterior.
- **Tablón de Anuncios:** esta opción nos permite acceder a los mensajes que se han publicado en la comunidad y mostrará una tabla con todos los mensajes, junto con su fecha. Si **usuario es un administrador**, aparecerá un botón en la parte inferior izquierda, debajo del tablón, con el texto “**Añadir mensaje**”, que desplegará un formulario que permitirá añadir un nuevo mensaje al tablón de anuncios.
- **Documentos:** pulsando en esta opción mostrará los **documentos relativos a la comunidad**, como pueden ser actas de reuniones, facturas, etc. Los documentos se mostrarán

como un conjunto de iconos con un nombre debajo. Tanto el icono como el nombre deberán ser un enlace que al pulsarlo permita a los usuarios descargar el documento en cuestión.

En el caso de que el **usuario sea un administrador**, debe aparecer un botón con el texto "**Añadir Documento**" que desplegará un formulario y le permitirá añadir nuevos documentos al directorio.

- **Incidencias:** aquí nos permitirá crear y manejar las incidencias que hemos creado mostrando una **lista con todas las incidencias** de la comunidad, junto con su fecha de creación y su estado. Para un **usuario**, solo se mostrarán **sus incidencias** y todas aquellas que hayan sido **marcadas como públicas** por los usuarios que las hayan creado. Para el administrador, serán visibles todas las incidencias.

Debajo de la lista de incidencias, debe haber un **botón** con el texto "**Crear incidencia**", que desplegará un formulario y permitirá a cualquier usuario crear una nueva incidencia. Además, si el **usuario conectado es el administrador**, aparecerán 2 botones a la derecha de cada incidencia, que permitirán tanto **borrar las incidencias** como **cambiar su estado**. En caso de que el usuario no sea administrador, solo aparecerá el botón "Borrar" en las incidencias que hayan sido creadas por el usuario.

- **Espacios Comunes:** se mostrarán los diferentes espacios comunes de la comunidad y se permitirá su reserva. Se mostrará una lista con todos los espacios comunes que hay en la comunidad, que permitirá clickar en ellos y acceder a la página de cada espacio común. Además, para los administradores, se mostrará un botón con el texto "Añadir Espacio" que mostrará un formulario que permitirá añadir un nuevo espacio común, con información como el nombre del espacio común, sus características y el horario de uso, además de la posibilidad de incluir una foto de dicho espacio.

- ◇ **Detalle de Espacio Común:** se muestran los **detalles de una zona común**. Aparecerá **una imagen**, si se hubiera incluido. Si no se cargara una imagen por defecto. Encima aparecerá el **nombre** del espacio común y al lado derecho de la imagen, una **descripción** de dicho espacio. Debajo de la descripción aparecerá el horario de uso de dicho espacio.

Debajo de este bloque, se mostrará un **calendario**. Los **días en los que no se pueda reservar** tendrán un fondo **rojo** y en los que aún **si se pueda** tendrán un fondo **verde**. Al **pinchar** sobre uno de estos días se desplegará una **lista con los posibles horarios** en los que se puede reservar, permitiendo pinchar en uno y pulsar el **botón Reservar** que habrá también aparecido junto con los horarios.

- **Finanzas:** se mostrará una **tabla con las finanzas** de la comunidad. Indicando los ingresos y el tipo de estos, como cuotas, seguros, etc., así como los gastos y su tipo, limpieza del edificio, arreglos, seguros, etc. La información de mostrará para este ejercicio, realizando una estimación, y se irá modificando de forma automática cuando se añadan gastos o ingresos.

Solo los **administradores** puede modificar esta tabla, mediante un formulario para añadir un registro en el que se podrá seleccionar el tipo de registro, si es un ingreso o gasto, y el tipo de cada uno, así como la cantidad. Cuando se agregue un nuevo registro, el sistema calculará el balance con los nuevos datos y los mostrará. Este formulario se mostrará después de pulsar el botón "Añadir" debajo de la tabla de finanzas.

- **Administrar:** esta opción solo es elegible por los usuarios con **rol de administrador** y

permite al administrador **gestionar** todos los aspectos de una **comunidad** en una misma pantalla de forma rápida.

En la zona de contenido se mostrará **información de al comunidad**, que será editable, así como los **elementos de todos** y cada uno de los **apartados** mencionados en este punto: Incidencias, Documentos, Finanzas, etc., con opciones para editar, consultar añadir o eliminar elementos a cada apartado. Los apartados se mostrarán uno debajo de otro con su correspondiente título y los botones adecuados para eliminar, editar o borrar elementos.

- **Barra de Búsqueda:** cabe destacar que **todas las secciones** incluirán, encima de la tabla o lista donde se muestra la información, una **barra de búsqueda** que permitirá seleccionar solo las entradas coincidentes con los términos que queramos.

- **Página de Añadir Comunidad:** esta pantalla, a la que solo tienen acceso los usuarios con rol administrador, permite **añadir una nueva comunidad** de vecinos. La interfaz mostrará un **formulario web** con diferentes datos que deberán rellenarse para dar de alta la comunidad, principalmente la **dirección**, que será única para cada comunidad. En el caso de que se pueda dar de alta la comunidad se mostrará un mensaje de éxito y se redireccionará a la página principal de comunidad. En caso contrario, se mostrará los mensajes de error adecuados.

- **Página de Perfil:** en esta pantalla, que se accede desde el menú vertical del icono de perfil, se mostrará un **formulario con toda la información del usuario** en la zona de contenido. Este formulario se podrá **editar** y actualizar mediante un botón que habrá debajo de él con el texto "Guardar".

Además, debajo de este aparecerá un **botón** con la opción "**Eliminar usuario**" que permitirá a un usuario eliminar su información, y que al ser pulsado ocultará el formulario con la información y mostrará una ventana de confirmación. Si la confirmación es positiva, se eliminará al usuario, y se redireccionará la pantalla principal.

Además de estas páginas, hay que tener en cuenta que la aplicación debe **visualizarse correctamente en cualquier dispositivo**. Aunque se añadirán restricciones más adelante en este aspecto, hay que tener en cuenta que algunos **elementos cambiarán** cuando la aplicación se visualice en **dispositivos pequeños**, como móviles o tablets. Estos elementos son:

- **Barra Superior:** en la barra superior todas las opciones del menú se agruparán en un **menú hamburguesa**, el cual se situará en la parte derecha de la barra.
- **Menú Lateral:** el menú lateral se ocultará, mostrando la interfaz una flecha en la parte superior izquierda, donde debería estar el menú. Al pulsar esta flecha

Más adelante, en la sección de adecuación al entorno y restricciones de diseño se proporcionará más información sobre el desempeño de la aplicación en dispositivos móviles y tablets.

## Glosario

- SSL** El protocolo SSL ó Secure Socket Layer es un protocolo de cifrado para comunicaciones en red que proporciona un canal seguro entre dos computadores o dispositivos que se comunican a través de internet o una red local.. 6
- HTTP** El protocolo HTTP o HyperText Transfer Protocol es un protocolo desarrollado por Tim Berners-Lee en 1989 y que permite la transferencia de archivos entre dos computadores o dispositivos y que es la base de comunicación en la World Wide Web.. 7
- idempotente** Un método HTTP es idempotente si el efecto que se produce en el servidor es el mismo si se manda una sola solicitud que si se mandan varias solicitudes iguales.. 7
- API** Una API o Application Programming Interface, es una pieza de código que permite a dos aplicaciones comunicarse entre sí para compartir información y funcionalidades.. 7
- frontend** En desarrollo web, el frontend es la parte de la aplicación que interactura con los usuarios, incluyendo todos los aspectos gráficos de la web y otros elementos que permiten navegar a los usuario dentro de la página.. 10
- backend** En desarrollo web, el backend es la parte de la aplicación que reside en el servidor y que implementa toda la lógica de la aplicación para permitir que la aplicación trabaje con la base de datos.. 10
- SQL** SQL o Structured Query Language es un lenguaje de programación diseñado para administrar y recuperar información de una base de datos, empleando el calculo relacional para hacerlo de una forma sencilla.. 10
- librerías** En informática. una librería es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, y que ofrece una interfaz bien definida para la funcionalidad que aporta.. 10
- ORM** Un ORM o Object Relational Mapping es una técnica que nos permite interactuar con bases de datos relacionales empleando funciones u objetos en vez de consultas SQL. También se conoce como ORM al software o librería que implementa esta funcionalidad.. 10
- hasing** El hashing es una herramienta criptográfica que se emplea para generar un conjunto de datos cifrados a partir de otro conjunto de datos, como una cadena o un archivo... 11
- DOS** Un ataque DOS o Denial of Service es un tipo de ataque que consiste en saturar de la conexión de una aplicación o ordenador de forma que este no pueda ofrecer lo servicios que estaba ofreciendo.. 11
- login** Proceso mediante el que se controla el acceso individual a un sistema informático mediante el uso de credenciales.. 12
- CTA** Un CTA, del inglés Call To Action, es una imagen o frase con un botón que busca persuadir a los usuarios para que realicen una acción concreta.. 12
- logout** Proceso mediante el que un usuario que tienen una sesión activa en un sistema informático puede cerrar ésta y salir del sistema.. 13
- menú hamburguesa** En diseño web un menú hamburguesa es un un tipo de menú que tiene forma de hamburguesa, consistiendo en un icono rectangular con varias líneas horizontales en su interior. Este menú suele emplearse para ahorrar espacio, especialmente en dispositivos móviles.. 15



## Referencias

- [1] Wikipedia - HTTP.  
<https://en.wikipedia.org/wiki/HTTP>.
- [2] Mozilla Developer Network - HTTP Methods.  
<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>.
- [3] Mozilla Developer Network - HTTP Methods: GET.  
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/GET>.
- [4] Mozilla Developer Network - HTTP Methods: POST.  
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/POST>.
- [5] Mozilla Developer Network - HTTP Methods: DELETE.  
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/DELETE>.
- [6] Mozilla Developer Network - HTTP Methods: PUT.  
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/PUT>.