

Snippetpedia

Gestor web de Snippets y Scripts de programación

Memoria de Proyecto Final de Máster
Máster Universitario en Ingeniería Informática
Desarrollo de aplicaciones web



Universitat
Oberta
de Catalunya

Autor: Carlos Alejandro González Ruiz

Consultor: Joan Giner Miguelez
Profesor: David García Solórzano

05/01/2021

Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Copyright

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual

Ficha del proyecto

Título del trabajo:	Snippetpedia – Gestor web de Snippets y Scripts
Nombre del autor:	Carlos Alejandro González Ruiz
Nombre del consultor/a:	Joan Giner Miguelez
Fecha de entrega:	01/2021
Titulación:	Máster Universitario en Ingeniería Informática
Área del Trabajo Final:	Desarrollo de aplicaciones web
Idioma del Trabajo:	Español
Palabras clave:	snippet script react net c# sqlserver web

Resumen del trabajo:

Cualquier desarrollador ha recurrido en algún momento de su vida profesional a diferentes páginas especializadas con el fin de encontrar una función, método o fragmento de código que resuelva un problema concreto.

Este tipo de pequeñas funciones auxiliares podrían ser necesitadas en otros proyectos por el mismo programador sin que éste, transcurrido cierto tiempo, sepa, cómo o dónde se encuentra dicho código, debiendo realizar la misma búsqueda de nuevo.

Snippetpedia trata de resolver un problema bastante común en el campo del desarrollo software, centralizando estos fragmentos de código en una plataforma cloud, y pudiendo compartirlos con otros usuarios, incorporando además un componente social, en el cual otros profesionales podrán sugerir mejoras o modificaciones de los snippets.

Para ello se implementará una solución con un cliente desarrollado en React, el cual consumirá una API Rest en .NET Core, que interactuará con una base de datos SQLServer. Toda la aplicación será alojada en Azure en forma de solución cloud.

Abstract:

Any developer has resorted at some point in his professional life to different specialized pages, in order to find a function, method or piece of code that solves a particular problem.

This kind of small auxiliary functions could be needed in other projects by the same developer without him knowing, after some time, how or where is this code, having to make the same research again.

Snippetpedia tries to solve a very common problem in the field of software development, centralizing these code fragments in a cloud platform, and being able to share them with other users, also incorporating a social component, in which other professionals will be able to suggest improvements or modifications to the snippets.

To do this, a solution will be implemented with a client developed in React, which will consume a Rest API in .NET Core, which will interact with a SQLServer database. The whole application will be hosted in Azure as a cloud solution.

Agradecimientos

Han sido muchas las personas que me han ayudado en este reto a lo largo de estos últimos años. Sin saberlo, pero con su apoyo, consejos y ayuda he conseguido llegar a la meta que hace un tiempo parecía inalcanzable y muy lejana. No dispongo de suficientes páginas para agradecerse a todas ellas. Pero sería también injusto dejar de nombrar a aquellos que particularmente han influido en que haya podido llegar al final de este camino.

Gracias Eider, por estar desde hace tantos años ahí, por seguir apoyándome, aún cuando muchas veces no te he podido dar el tiempo que te mereces.

Gracias Milo, por hacerme ver y entender las nuevas prioridades de mi vida.

Gracias Alex, por darme una idea que poder llevar a cabo, y una larga lista de buenos consejos.

Gracias Christian, por nuestras largas jornadas de *Brainstorming* en nuestra *Theke* favorita.

Gracias Juan, por nuestras provechosas y entretenidas sesiones de *“React for Dummies”*.

Gracias Puchy, Pollo y Alba, por la cena a la que os voy a tener que invitar.

Y por último y no menos importantes, gracias a los enanitos que tantas cosas han hecho por mí.

Índice

1	Introducción	11
1.1	Contexto y justificación del trabajo	11
1.2	Objetivos	15
1.2.1	Objetivos principales	15
1.2.2	Objetivos secundarios	16
1.3	Metodología	16
1.4	Planificación	16
1.4.1	Jornada laboral y duración del proyecto	16
1.4.2	Actividades e hitos	17
1.4.3	Diagrama de Gantt	19
1.4.4	Riesgos	20
1.4.4.1	Criterios	20
1.4.4.2	Estimación de riesgos	20
1.5	Tecnologías, plataformas y herramientas	21
1.5.1	Tecnologías y plataformas	21
1.5.2	Herramientas	21
1.6	Productos resultantes obtenidos	22
2	Análisis y definición de requisitos	24
2.1	Análisis del mercado	24
2.1.1	Lepton	24
2.1.2	Snippetslab	25
2.1.3	Dash	26
2.2	Requisitos	27
2.2.1	Requisitos funcionales	27
2.2.2	Requisitos no funcionales	28
3	Diseño y Arquitectura	30
3.1	Diseño centrado en el usuario (DCU)	30
3.1.1	Usuarios y contexto de uso	30
3.1.1.1	Entrevistas	30
3.1.1.2	Conclusiones	34
3.1.2	Flujos de interacción	35
3.1.3	Prototipado	36

3.1.3.1	Página de inicio	36
3.1.3.2	Registro de usuarios.....	37
3.1.3.3	Login	38
3.1.3.4	Reestablecer contraseña	39
3.1.3.5	Dashboard	40
3.1.3.6	Crear / modificar snippet.....	41
3.1.3.7	Seleccionar tags	42
3.1.3.8	Crear / modificar fichero	42
3.1.3.9	Perfil de usuario.....	43
3.2	Diseño técnico.....	44
3.2.1	Actores del sistema.....	44
3.2.2	Diagramas de casos de uso.....	44
3.2.2.1	Diagrama de paquetes de casos de uso	45
3.2.2.2	Diagrama de casos de uso: Usuarios.....	45
3.2.2.3	Diagrama de casos de uso: Búsquedas	46
3.2.2.4	Diagrama de casos de uso: Snippets	46
3.2.3	Listado de casos de uso	47
3.2.4	Arquitectura.....	58
3.2.4.1	Visión general	59
3.2.4.2	Modelo de datos	59
3.2.4.2.1	Diagrama.....	60
3.2.4.2.2	Tablas.....	60
3.2.5	Integración y entrega continua.....	65
4	Implementación	67
4.1	Visión general.....	67
4.2	Herramientas y librerías	68
4.2.1	Front-end.....	68
4.2.1.1	Librerías	68
4.2.1.2	Herramientas de desarrollo.....	71
4.2.2	Back-end	72
4.2.2.1	Librerías	73
4.2.2.2	Herramientas de desarrollo.....	75
4.3	Estructura del proyecto	76
4.3.1	Front-end.....	77
4.3.2	Back-end	78
4.4	Configuración cloud.....	79

4.5	API Rest	81
5	Demo	84
5.1	Acceso.....	84
5.2	Problemas conocidos	84
6	Pruebas	85
7	Instalación y configuración	86
8	Conclusiones	87
8.1	Objetivos	87
8.2	Enseñanzas adquiridas	87
8.2.1	Estimaciones.....	87
8.2.2	Automatización.....	88
8.3	Futuro	88
9	Bibliografía.....	90

Figuras y tablas

Índice de figuras

Figura 1: Estimación del número de desarrolladores software en el mundo	11
Figura 2: Encuesta de StackOverflow 2020 - El fenómeno "sensación de ya haber estado aquí"	12
Figura 3: Encuesta de StackOverflow 2020 - Reacción ante problemas	13
Figura 4: Ejemplo de un snippet en lenguaje de programación PHP	14
Figura 5: Diagrama de Gantt	19
Figura 6: Diagrama de flujos de interacción	36
Figura 7: Diagrama de paquetes de casos de uso	45
Figura 8: Diagrama de casos de uso: Usuarios	45
Figura 9: Diagrama de casos de uso: Búsquedas	46
Figura 10: Diagrama de casos de uso: Snippets	46
Figura 11: Arquitectura - Visión general	59
Figura 12: Diagrama de base de datos	60
Figura 13: Proceso de integración y entrega continua	65
Figura 14: Implementación - Visión general	67
Figura 15: Ejemplo de uso del editor Ace	69
Figura 16: Fichero de internacionalización del proyecto	69
Figura 17: Ejemplo de aplicación basada en Material-UI	70
Figura 18: Ejemplo del selector de color react-color	70
Figura 19: Ejemplo de notificación toast	71
Figura 20: Ejemplo de error con google chrome dev tools	72
Figura 21: Ejemplo de uso del patrón Mediator	73
Figura 22: Entorno de desarrollo Rider	75
Figura 23: Cliente SGBD Datagrip	76
Figura 24: Suscripción a Azure	79
Figura 25: Grupo de recursos en Azure	80
Figura 26: Costes Azure desde septiembre 2020	81
Figura 27: Pruebas unitarias en proyecto back-end	85

Índice de tablas

Tabla 1: Dedicación diaria y total al proyecto	17
Tabla 2: Planificación inicial del proyecto	18
Tabla 3: Criterios en la estimación de riesgos	20
Tabla 4: Estimación de riesgos	21
Tabla 5: Herramientas de desarrollo	22
Tabla 6: Servicios de desarrollo	22
Tabla 7: Requisitos funcionales	28
Tabla 8: Requisitos no funcionales	29

Tabla 9: Cuestionario general y específico DCU	34
Tabla 10: Prototipo página de inicio	37
Tabla 11: Prototipo registro de usuario	38
Tabla 12: Prototipo login de usuario.....	38
Tabla 13: Prototipo reestablecer contraseña	39
Tabla 14: Prototipo dashboard	41
Tabla 15: Prototipo creación y modificación de snippet	42
Tabla 16: Prototipo seleccionar tags	42
Tabla 17: Prototipo creación y modificación de fichero	43
Tabla 18: Prototipo perfil de usuario	44

1 Introducción

1.1 Contexto y justificación del trabajo

La tecnología ha avanzado enormemente en los últimos 20 años. Esto es perceptible en casi cualquier ámbito de nuestra vida: vemos la tele conectados a internet, conversamos con nuestros amigos a través del móvil, nuestros coches están conectados y nos guían a nuestro destino mediante aplicaciones, pedimos comida a domicilio a través de sitios web...

En todas estas actividades existe un nexo común: un desarrollador se ha encargado de convertir una serie de requisitos funcionales en algo intangible pero que puede ser empleado por un usuario final.

Según un estudio realizado por la web statista.com, el número estimado de programadores en 2018 era de 23 millones en todo el mundo. En 2024 se espera que este número alcance casi los 29 millones, siendo uno de los empleos más demandados a nivel global por las empresas.

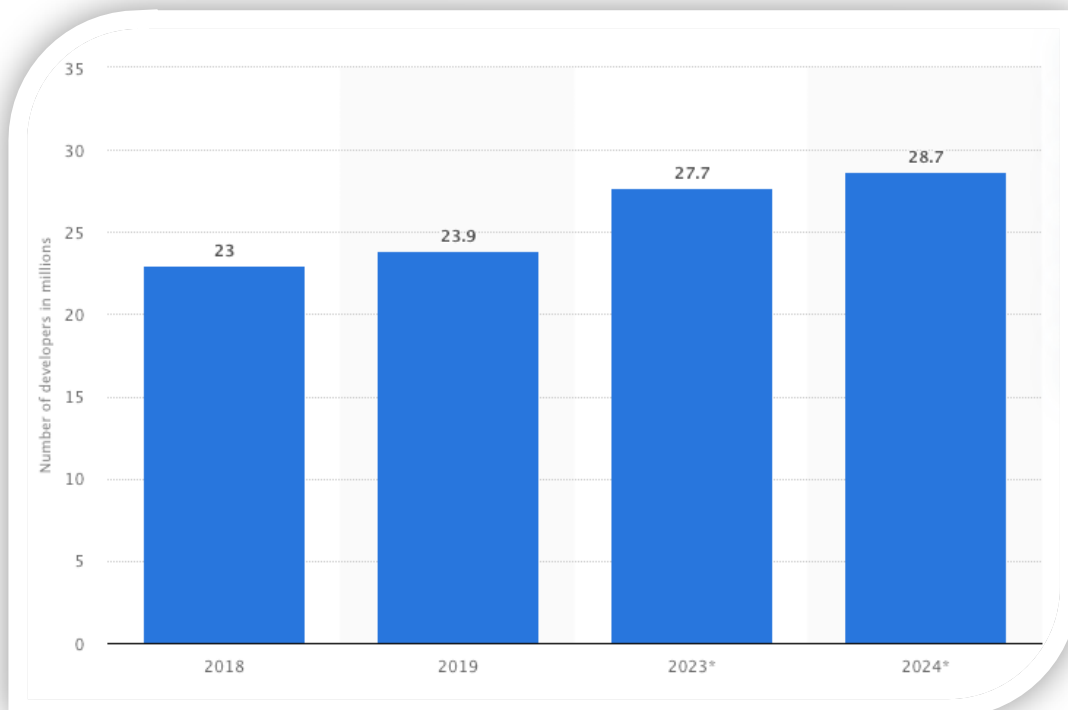


Figura 1: Estimación del número de desarrolladores software en el mundo¹

¹ Análisis realizado por el sitio web statista.com: <https://www.statista.com/statistics/627312/worldwide-developer-population/>

Los desarrolladores, a lo largo de su vida profesional adquieren experiencia, nuevos conocimientos, hábitos y formas de programar. Todo programador es consciente de que el éxito de su carrera depende de este aprendizaje continuo. Pero a lo largo de los diferentes proyectos y diferentes empresas en las que éste desarrolla su labor profesional, se requieren de forma recurrente ciertas funcionalidades que ya han sido desarrolladas en el pasado.

Este sentir generalizado en cualquier programador con un bagaje anterior significativo, se recoge perfectamente en la encuesta anual de StackOverflow², en la cual se planteaba la cuestión relativa a la reacción de los usuarios cuando se dan cuenta de que el enlace del sitio web donde se describe una solución a un problema actual, ha sido ya visitado en otras ocasiones.

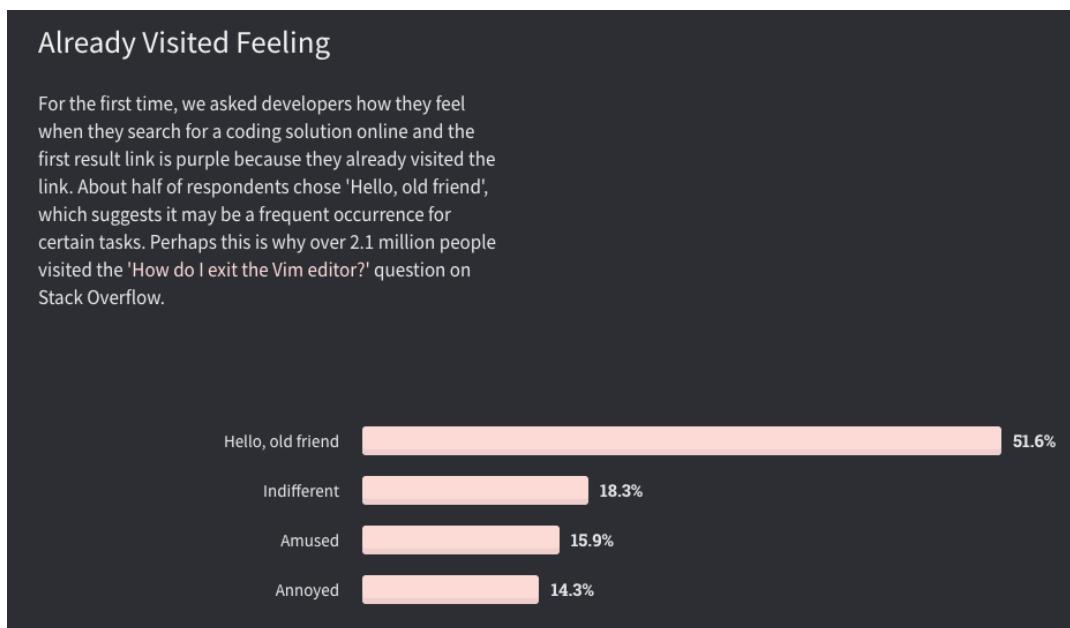


Figura 2: Encuesta de StackOverflow 2020 - El fenómeno "sensación de ya haber estado aquí"

Esto no es para nada ajeno al autor de este trabajo de fin de máster, el cual ha experimentado en numerosas ocasiones esta sensación. Con el fin de ofrecer una solución que permita apoyar a los programadores con esta problemática, se plantea Snippetpedia como un sistema web accesible en cualquier momento por cualquier dispositivo, en el cual un programador podrá disponer y gestionar el conocimiento propio de aquellos recursos relacionados con el desarrollo software, que, de forma recurrente deben ser empleados y que sin embargo y debido a la naturaleza o complejidad de los mismos, muchas veces son sólo localizables tras una nueva búsqueda en internet.

² La encuesta anual es accesible a través del siguiente enlace: <https://insights.stackoverflow.com/survey/2020>

La propia encuesta de StackOverflow recoge también la reacción habitual de un usuario cuando no encuentra solución a un problema específico:

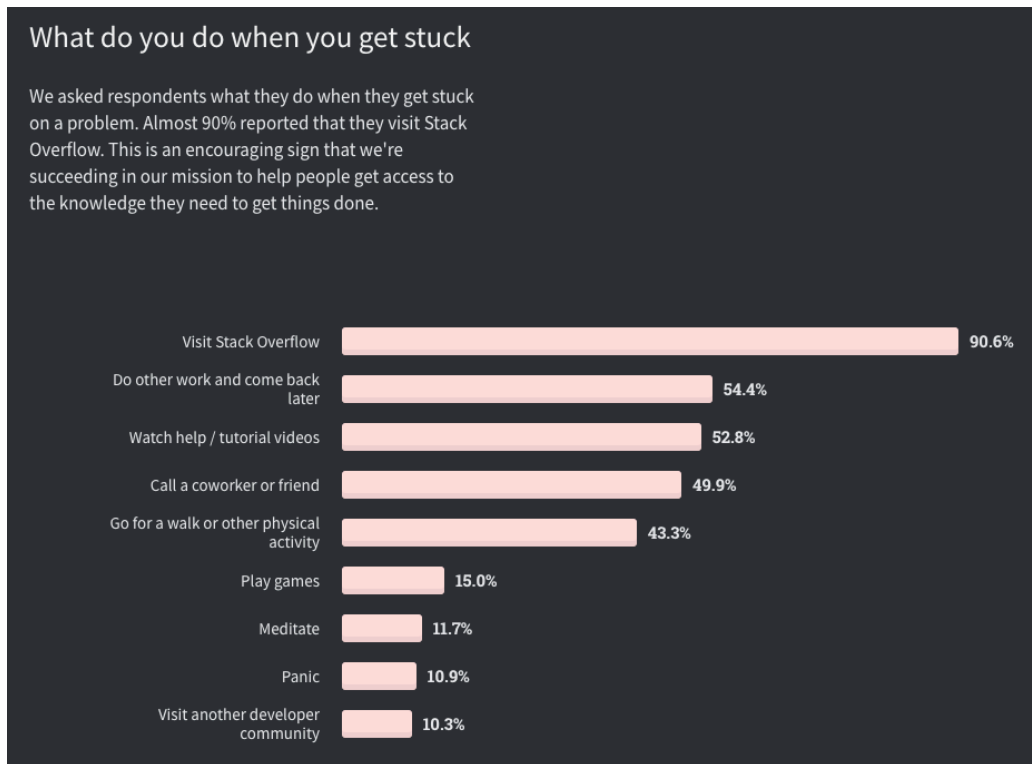


Figura 3: Encuesta de StackOverflow 2020 - Reacción ante problemas

Como se puede observar a tenor de las respuestas a ambas preguntas, el programador medio acabará buscando de forma recurrente en StackOverflow una y otra vez una solución al mismo problema en diferentes momentos de su carrera profesional.

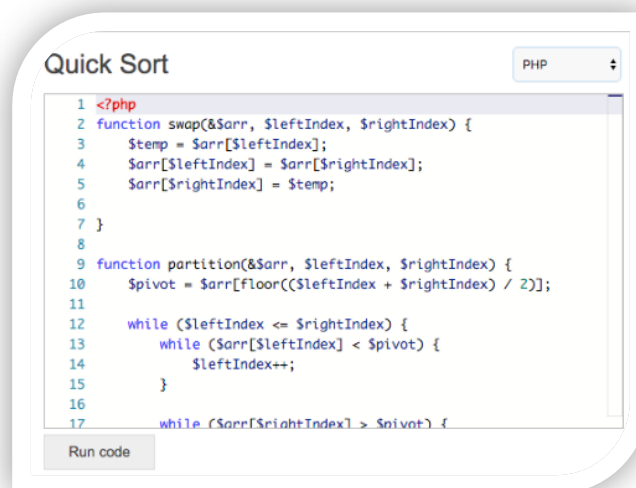
En el mercado actual no existen muchas alternativas software que aborden esta problemática, y las que lo hacen, son en su mayoría aplicaciones de escritorio en las cuales la información se almacena de forma local. Pero ¿qué ocurre cuando el programador está trabajando en otro equipo y no ha instalado dicho software?

Por lo expuesto en los puntos anteriores se puede deducir que los programadores en los diferentes proyectos en los que trabajen reutilizarán en mayor o menor medida ciertos fragmentos de código, scripts o recursos textuales, muchas veces sin ser conscientes de ello. La encuesta de la Figura 2: Encuesta de StackOverflow 2020 - El fenómeno "sensación de ya haber estado aquí", así lo evidencia.

Con el fin de servir como herramienta de ayuda y gestión del conocimiento para los programadores, y en el marco de la asignatura del trabajo de fin de máster universitario de la Universitat Oberta de Catalunya (UOC), se presenta el proyecto Snippetpedia.

Snippetpedia es un sistema o aplicación web, que trata de centralizar, estructurar y agrupar el conocimiento de cada programador en su propio espacio personal, en el cuál éste podrá organizar sus *snippets* y los ficheros que de éste dependan, de una forma sencilla y fácilmente localizable.

Un *snippet* es un término en inglés utilizado en el desarrollo software para referirse a pequeños fragmentos reusables de código fuente o texto. Comúnmente son definidas como unidades o métodos funcionales que pueden ser integrado en diferentes módulos o aplicaciones de mayor entidad, aportando una funcionalidad concreta.



```
1 <?php
2 function swap(&$arr, $leftIndex, $rightIndex) {
3     $temp = $arr[$leftIndex];
4     $arr[$leftIndex] = $arr[$rightIndex];
5     $arr[$rightIndex] = $temp;
6
7 }
8
9 function partition(&$arr, $leftIndex, $rightIndex) {
10     $pivot = $arr[floor(($leftIndex + $rightIndex) / 2)];
11
12     while ($leftIndex <= $rightIndex) {
13         while ($arr[$leftIndex] < $pivot) {
14             $leftIndex++;
15         }
16
17         while ($arr[$rightIndex] > $pivot) {
```

Figura 4: Ejemplo de un snippet en lenguaje de programación PHP

El sistema ofrecerá en su versión final las siguientes funcionalidades, las cuales serán detalladas de forma más específica en capítulos posteriores:

- **Usuarios:** los usuarios podrán registrarse en la plataforma y tras la activación podrán acceder al sistema. Adicionalmente, y como cualquier sistema web convencional, el usuario podrá administrar su perfil y, en caso de ser necesario, reestablecer su contraseña.
- **Gestión de etiquetas personales:** la creación de snippets y cuando se alcanza un número importante de éstos, puede dificultar su localización. Con el fin de optimizar la búsqueda, se ofrece un sistema de etiquetas o tags, que permitan categorizarlos.

- **Gestión de snippets:** los usuarios del sistema podrán crear sus propios contenidos, tanto de forma pública como privada. Asimismo, será posible mediante una búsqueda global, localizar snippets de otros usuarios que los hayan creado de forma pública. Éstos además podrán ser añadidos a una lista personal de favoritos con el fin de facilitar su localización.
- **Gestión de ficheros:** un snippet se puede componer de un número indeterminado de ficheros en diferentes lenguajes de programación. Snippetpedia permitirá además de la descarga y gestión de dichos ficheros, otras funciones avanzadas en materia de gestión de los contenidos, basándose para ello en editor avanzado que ofrezca funcionalidades de auto completado.
- **Interacción de usuarios:** con el fin de favorecer la interacción de los usuarios y el aprendizaje común, se ofrecerá la posibilidad de que otros usuarios comenten snippets que han sido creados de forma pública. Este sistema de retroalimentación beneficiará a los propios usuarios en tanto y cuanto podrán recibir propuestas de mejora de sus contenidos o podrán ayudarse entre ellos.

1.2 Objetivos

A continuación, se describen los objetivos deseados con la realización de este proyecto.

1.2.1 Objetivos principales

Objetivos de proyecto:

- Implementar un servicio online basado en cloud de almacenamiento y búsqueda de snippets
- Facilitar la búsqueda de snippets ofreciendo filtros de búsqueda adecuados
- Fomentar una retroalimentación entre usuarios que permita valorar y comentar los fragmentos de códigos proporcionados por los usuarios

Objetivos para el usuario final:

- Interfaz sencilla e intuitiva
- Búsqueda eficiente y con tiempos de respuesta mínimos y resultados acordes a los términos introducidos

Objetivos personales del autor del TFM:

- Adquirir nuevos conocimientos en una tecnología hasta ahora desconocida para el autor como es React.
- Mejorar los conocimientos existentes con SQLServer y Azure.
- Adquirir la experiencia necesaria en la elaboración e implementación de un software siguiendo los estándares actuales y las recomendaciones de los profesores.

1.2.2 Objetivos secundarios

Adicionalmente se persiguen los siguientes objetivos:

- Llegar a convertir la aplicación en un producto, el cual, una vez finalizada la primera versión, pueda ser empleado de forma profesional por cualquier usuario.
- Dado que para el desarrollo de este proyecto se emplean tecnologías estandarizadas como REST, se podrán realizar plugins o integraciones para cada uno de los entornos de desarrollo (p.ej: Visual Studio Code, Eclipse, etc...) de forma que se pueda acceder dentro del propio IDE³ a la lista de snippets del usuario.

1.3 Metodología

En la fase de concepción y desarrollo del proyecto, y debido al reducido número de integrantes del equipo, se ha seguido una metodología secuencial iterativa, comenzando con una fase inicial de concepción a nivel de requisitos funcionales y técnicos. Tras esta definición, comenzará una fase de diseño, mas cercana a lo que será el producto resultante, finalizando con una fase de implementación y pruebas.

En el desarrollo se ha tratado además de incorporar elementos y herramientas provenientes de las metodologías ágiles, haciendo uso por ejemplo de herramientas como Jira para la definición de los diferentes casos de uso, así como el control de las diferentes tareas resultantes mediante paneles Kanban.

1.4 Planificación

En el presente capítulo se describe la planificación general del proyecto tomando como referencia la disponibilidad y posible dedicación diaria.

1.4.1 Jornada laboral y duración del proyecto

Día de la semana	Inicio / Fin	Horas
Lunes	19:00 – 21:30	2,5
Martes	19:00 – 21:30	2,5
Miércoles	19:00 – 21:30	2,5
Jueves	19:00 – 21:30	2,5
Viernes	18:00 – 21:30	3,5
Sábado	08:00 – 15:00	7
Domingo	08:00 – 15:00	7
Excepciones	25/12, 26/12, 31/12	-

³ Integrated development environment o entorno de desarrollo integrado

TOTAL SEMANAL	-	30
TOTAL PROYECTO	16/09/2020 – 15/01/2021	420,5

Tabla 1: Dedicación diaria y total al proyecto

1.4.2 Actividades e hitos

Hito	Actividad	Duración	Inicio	Fin
Entrega 1 (Plan de trabajo)	PEC 1	55h	16/09/20	29/09/20
	Análisis de tecnologías	7h	16/09/20	18/09/20
	Formación React	20h	19/09/20	23/09/20
	Búsqueda de recursos bibliográficos	8h	24/09/20	26/09/20
	Redacción documento PEC1	20h	26/09/20	29/09/20
Entrega 2 (Diseño)	PEC 2	103,5h	30/09/20	28/10/20
	Configuración de los entornos y herramientas	15h	30/09/20	03/10/20
	Diseño conceptual	25h	03/10/20	07/10/20
	Prototipado de bajo y alto nivel	40h	07/10/20	16/10/20
	Diseño de la arquitectura de sistema	3,5h	16/10/20	17/10/20
	Diseño de la base de datos	5h	17/10/20	20/10/20
	Definición del <i>walking sekeleton</i> ⁴ en cliente y servicio	10h	21/10/20	24/10/20
	Redacción documento PEC2	5h	24/10/20	28/10/20
Entrega 3 (Implementación)	PEC 3	164h	29/10/20	09/12/20
	Definición API	10h	29/10/20	31/10/20
	Implementación funcionalidades de servicio	40h	31/10/20	10/11/20
	Implementación funcionalidades básicas de cliente	104h	10/11/20	05/12/20
	Elaboración vídeo primera versión	5h	06/12/20	07/12/20
	Redacción documento PEC3	5h	08/12/20	09/12/20
Entrega final	Entrega final	84,5h	10/12/20	05/01/21
	Implementación funcionalidades restantes de cliente	44,5h	10/12/20	24/12/20
	Bugfixing	5h	24/12/20	25/12/20
	Desarrollo tests unitarios	15h	25/12/20	28/12/20

⁴ En desarrollo software, un walking skeleton describe la implementación más simple de un software funcional. Walking hace referencia a que la mínima implementación ya es ejecutable.

	Elaboración vídeo versión final	10h	28/12/20	01/01/21
	Redacción documento entrega final	10h	01/01/21	05/01/21
Defensa	Defensa virtual	13,5h	11/01/21	15/01/21
	Defensa virtual	13,5	11/01/21	15/01/21

Tabla 2: Planificación inicial del proyecto

1.4.3 Diagrama de Gantt

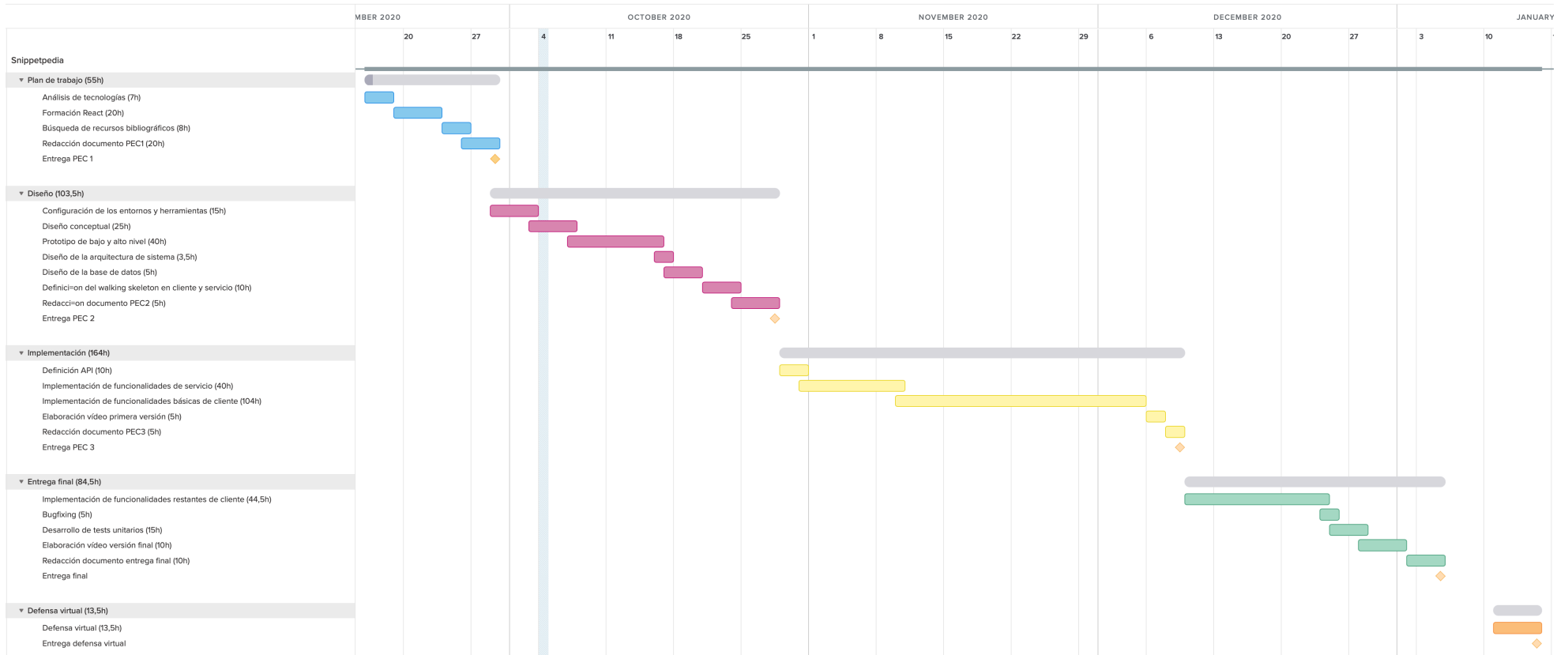


Figura 5: Diagrama de Gantt⁵

⁵ Diagrama elaborado con Teamgantt. Más información: <https://www.teamgantt.com/>

1.4.4 Riesgos

Para la confección de la lista de riesgos en la planificación, se han tenido en cuenta tres atributos, que ayudarán a entender la criticidad de las posibles desviaciones del proyecto.

1.4.4.1 Criterios

En la confección de la lista de riesgos en la planificación, se han tenido en cuenta tres atributos, que ayudarán a entender la criticidad de las posibles desviaciones del proyecto.

Atributo	Valor	Descripción
Impacto	Catastrófico	Pérdida del sistema
	Crítico	Impacto severo
	Marginal	Impacto menor
Probabilidad	Muy probable	> 70%
	Probable	Entre 30% y 70%
	Improbable	< 30%
Desviación	Mayor	> 1 semana
	Media	1 semana
	Menor	1-2 días

Tabla 3: Criterios en la estimación de riesgos

1.4.4.2 Estimación de riesgos

Riesgo	Descripción	Imp.	Prob.	Desv.
Estimación componente cliente	La definición de la planificación para el componente cliente no está equilibrado	Crítico	Probable	Mayor
Estimación componente servicio	La definición de la planificación para el componente servicio no está equilibrado	Marginal	Improbable	Menor
Problemas configuración o rendimiento infraestructura cloud	La configuración de la infraestructura cloud es más compleja de lo esperado o el rendimiento no es óptimo	Marginal	Improbable	Menor

Sobrecoste configuración cloud	La inexperiencia en entornos cloud puede provocar que se incurran en costes adicionales no esperados	Marginal	Improbable	Menor
Disponibilidad debido a razones laborales	Por razones de ámbito laboral es posible que en algún momento no se disponga del tiempo suficiente diario definido en Jornada laboral y duración del proyecto1.4.1	Marginal	Probable	Media
Exceso de funciones	El número de funciones o la complejidad de estas excede el tiempo asignado en total para el desarrollo completo del proyecto	Crítico	Improbable	Media
Usabilidad	El diseño final no es intuitivo o existen problemas de usabilidad	Crítico	Improbable	Media

Tabla 4: Estimación de riesgos

1.5 Tecnologías, plataformas y herramientas

1.5.1 Tecnologías y plataformas

El sistema a desarrollar, al tratarse de una aplicación web, podrá ser ejecutado en cualquier plataforma que disponga de un navegador web que soporte las tecnologías ya de facto estandarizadas como HTML5 o CSS3.

1.5.2 Herramientas

A continuación, se enumeran las herramientas de desarrollo empleadas en la implementación de la solución final. Dicho software dispone de versiones para Windows y Mac y deberá ser instalado con el fin de poder ejecutar la solución de forma local.

Herramienta	Descripción	Licencia	Enlace
Visual Studio Code	Entorno de desarrollo integrado (IDE) necesario para el desarrollo de la parte de cliente de la aplicación con React, Typescript, HTML y CSS	MIT	Enlace

Rider	Entorno de desarrollo integrado (IDE) necesario para el desarrollo de la parte de servicio de la aplicación con .NET Core	Comercial (Permitido en contexto universitario)	Enlace
Datagrip	Cliente que permite gestionar y consultar la mayor parte de los proveedores de bases de datos actuales	Comercial (Permitido en contexto universitario)	Enlace
SQLServer	Sistema de gestión de bases de datos relacional	Comercial (Permitido en contexto universitario)	Enlace

Tabla 5: Herramientas de desarrollo

En la siguiente tabla se describe la lista de servicios empleados en el proyecto.

Herramienta	Descripción	Licencia
Jira	Herramienta web para la gestión administrativa de los diferentes casos de uso y tareas. Permite la generación automática de las planificaciones y visualización de las tareas y user stories en forma de tableros Kanban	Comercial / Gratuita
Github	Sistema de control y gestión de código fuente basado en Git. Adicionalmente se hará uso de funciones avanzadas CI/CD que permitan verificar el código subido y automatizar los despliegues de la parte de cliente y servicio	Gratuito
Azure	Servicio cloud en el que se alojarán los componentes de cliente y servicio, así como la base de datos	Comercial (100\$ y un año de servicios gratuitos en contexto universitario)

Tabla 6: Servicios de desarrollo

1.6 Productos resultantes obtenidos

Los productos resultantes tras la finalización de este proyecto serán los siguientes:

- Aplicación cliente en React
- Aplicación backend en .NET Core

- Documento de instalación y configuración de la aplicación
- Código fuente
- Memoria (documento actual) que describe todas las fases del proyecto, así como los requisitos funcionales y técnicos
- Presentación: video explicativo de los aspectos más importantes del sistema
- Repositorio GitHub público: <https://github.com/pumbinha/snippetpedia>
- Sitio web de acceso público: <https://www.snippetpedia.com>

2 Análisis y definición de requisitos

2.1 Análisis del mercado

En este capítulo se llevará a cabo un análisis de los gestores de snippets más populares, independientemente de la plataforma, para finalizar con un análisis comparativo entre los mismos. Dicho análisis ha servido como punto de partida y estudio inicial para definir las características que el sistema final debería ofrecer.

2.1.1 Lepton

Lepton	
Sitio Web	http://hackjutsu.com/Lepton/
Versión	1.9.1
Plataforma	Windows, Linux, MacOS
Licencia	MIT
Precio	Gratuito
Descripción	<p>Lepton es un gestor de snippets de código abierto desarrollado en Electron⁶ y que basa su funcionamiento en los Gists⁷ de Github. Entre sus funcionalidades más destacadas se encuentran:</p> <ul style="list-style-type: none"> - Snippets públicos y privados basados en Gists - Tags ilimitados - Markdown o resaltado según lenguaje - Soporte multiplataforma - Búsqueda - Dashboard
Imágenes	

⁶ Framework de código abierto para aplicaciones de escritorio utilizando tecnologías web desarrollado y mantenido por GitHub. Más información: <https://www.electronjs.org/>

⁷ Sistema propio de Github basado en git para el almacenamiento de snippets. Más información: <https://docs.github.com/en/free-pro-team@latest/github/writing-on-github/creating-gists - about-gists>



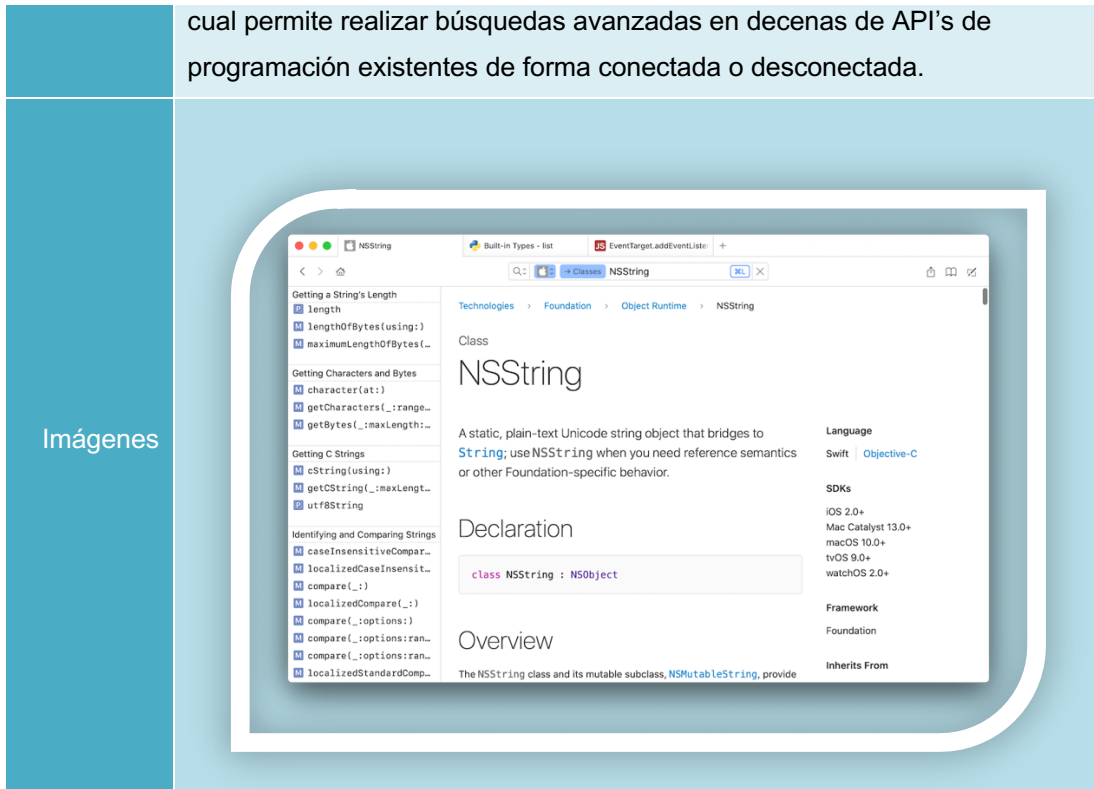
2.1.2 Snippetslab

Lepton	
Sitio Web	https://www.renfei.org/snippets-lab/
Versión	1.9.2
Plataforma	MacOS

Licencia	Comercial
Precio	10,99 €
Descripción	Snippetslab es una aplicación de escritorio dirigida a dispositivos Mac que facilita la gestión de los diferentes contenidos con una agrupación jerárquica basada en carpetas, accesos directos, etiquetas y los llamados Smart Groups (grupos inteligentes), que permite una búsqueda avanzada basada en diferentes criterios como fechas de creación y modificación.
Imágenes	

2.1.3 Dash

Lepton	
Sitio Web	https://kapeli.com/dash
Versión	5.5.1
Plataforma	MacOS
Licencia	Comercial
Precio	29,99 €
Descripción	Dash es una aplicación de escritorio que permite no sólo la gestión de snippets personales, sino una plena integración con múltiples entornos de desarrollo con el fin de agilizar el proceso de desarrollo. Entre otras funciones interesantes se encuentra un navegador de documentación, el



2.2 Requisitos

A continuación, se enumeran los diferentes requisitos funcionales y no funcionales de la aplicación, entendiéndose como valor relativo a la prioridad un valor numérico entre 1 y 5 (1 mayor prioridad, 5 menor prioridad).

2.2.1 Requisitos funcionales

Código	Descripción
SNIP-RF-1	La aplicación permitirá al usuario iniciar sesión con un correo electrónico y una contraseña.
SNIP-RF-2	Un usuario podrá recuperar su contraseña si la ha olvidado.
SNIP-RF-3	Será posible crear una nueva cuenta en el sistema introduciendo una serie de datos básicos: alias, correo electrónico, contraseña. Tras la confirmación de la cuenta de correo, vía link de activación, el usuario podrá acceder al sistema.
SNIP-RF-4	Un usuario podrá modificar su propio perfil.
SNIP-RF-5	Un usuario podrá consultar la información de otros usuarios: tan sólo será visible información básica del mismo: alias, imagen, notas de usuario.

SNIP-RF-6	Será posible realizar búsquedas generales de snippets mediante el uso de palabras claves asociadas al título, descripción y lenguaje del snippet.
SNIP-RF-7	Un usuario podrá crear listas de snippets. Para ello será necesario un nombre y una descripción.
SNIP-RF-8	Un usuario podrá modificar su propia lista de snippets.
SNIP-RF-9	Un usuario podrá eliminar su propia lista de snippets, si ésta no contiene ningún snippet.
SNIP-RF-10	Un usuario podrá dentro de una lista filtrar los snippets que ésta contiene
SNIP-RF-11	Un usuario podrá añadir un nuevo snippet a una lista ya existente
SNIP-RF-12	Un usuario podrá modificar un snippet propio existente de una lista
SNIP-RF-13	Un usuario podrá eliminar un snippet propio existente de una lista
SNIP-RF-14	Un usuario podrá añadir un snippet propio o ajeno a la lista de favoritos
SNIP-RF-15	Un usuario podrá valorar positiva o negativamente un snippet ya existente
SNIP-RF-16	Un usuario podrá hacer un fork (copiar como snippet propio) de un snippet de otro usuario
SNIP-RF-17	Un usuario podrá añadir uno o varios comentarios a un snippet existente

Tabla 7: Requisitos funcionales

2.2.2 Requisitos no funcionales

Código	Descripción	Prioridad
SNIP-RNF-1	La aplicación podrá ser utilizada empleando navegadores web estándar actualmente aceptados (Edge, Firefox, Chrome)	1
SNIP-RNF-2	La aplicación deberá cumplir con los estándares marcados por el W3C (HTML 5.0 o superior, CSS 3.0)	3
SNIP-RNF-3	El sitio web deberá tener una estructura clara y ordenada del contenido, agrupando las funciones existentes de forma visible e intuitiva para el usuario final	2
SNIP-RNF-4	El sitio web cargará los diferentes contenidos de forma asíncrona e informativa, de forma que no se interrumpa la experiencia final del usuario	4
SNIP-RNF-5	En los formularios de entrada y, en la medida de lo posible, se tratará siempre de informar al usuario cuando los valores introducidos no sean correctos	3
SNIP-RNF-6	La aplicación deberá ser escalable vertical y horizontalmente con el fin de prevenir deficiencias en el servicio	1

SNIP-RNF-7	Las peticiones concurrentes deberán siempre reflejar estados consistentes en los datos mostrados	1
SNIP-RNF-8	Los datos deberán ser respaldados cada 24 horas. Éstos serán almacenados de forma segura.	1
SNIP-RNF-9	La aplicación hará uso de un diseño <i>responsive</i> ⁸	2
SNIP-RNF-10	La disponibilidad de la aplicación de como mínimo un 99%	2

Tabla 8: Requisitos no funcionales

⁸ Técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos con distintas resoluciones.

3 Diseño y Arquitectura

3.1 Diseño centrado en el usuario (DCU)

En el diseño centrado en el usuario, el enfoque se centra en el éste a través de la información sobre las necesidades y objetivos que mediante diferentes técnicas de investigación se obtienen del mismo.

3.1.1 Usuarios y contexto de uso

Con el fin de identificar los diferentes perfiles de usuario y contextos de uso, se ha decidido emplear un método de indagación con usuarios de la ingeniería de la usabilidad, que permitirá reconocer patrones y costumbres existentes en un grupo de trabajo muy específico como son los desarrolladores de software: la dinámica de grupo.

Mediante esta técnica y a través de la realización de entrevistas grupales con programadores con distinta experiencia en el ámbito del desarrollo software, se trata de identificar las reacciones de cada uno ante una serie de problemas propuestos. Es importante resaltar que el uso de esta metodología tiene un fin creativo en el cual se trata de detectar aquellos elementos que deberían ser priorizados y que, en definitiva, hagan que el producto final pueda resultar interesante para una amplia mayoría.

3.1.1.1 Entrevistas

Las entrevistas grupales se dividieron en dos fases:

- En la primera fase se plantearon una serie de escenarios comunes en el desarrollo software con el fin de obtener respuestas y conductas habituales entre los programadores.
- En la segunda fase se describe el concepto inicial del sistema a desarrollar. A continuación, se plantean una serie de preguntas relativas al mismo con el fin de detectar puntos positivos y negativos del mismo, así como posibles funciones ocultas que pudiesen llegar a ser interesantes.

Con el fin de obtener la mayor variedad de respuestas posibles, se ha buscado un grupo heterogéneo de programadores con distinta experiencia, la cual en el ámbito del desarrollo software, oscila entre 1 y 15 años.

A continuación, se presenta el grupo de individuos que formaron parte de la entrevista:

- **Luis González:**
 - o Edad: 36
 - o Profesión: Comercial
 - o Tecnologías: C++

- Experiencia: Aunque ya no se dedica profesionalmente al desarrollo, Luis ha trabajado durante su formación reglada con C++ y HTML.
- **Alberto Bolsa:**
 - Edad: 34
 - Profesión: Desarrollador Fullstack
 - Tecnologías: Alberto comenzó tras sus estudios en Ingeniería informática su carrera profesional en el mundo del desarrollo, realizando su actividad durante los últimos 15 años en distintas empresas de diverso tamaño. Se ha especializado en tecnologías del ecosistema Microsoft y actualmente realiza funciones como programador Fullstack en una empresa del sector bancario.
 - Experiencia: 15 años
- **David Salort:**
 - Edad: 34
 - Profesión: Desarrollador aplicaciones móviles
 - Tecnologías: Kotlin, Android, Java
 - Experiencia: David trabaja desde hace 4 años desarrollado aplicaciones móviles destinadas a dispositivos móviles, concretamente aquellos basados en Android.
- **Alba Sanz:**
 - Edad: 36
 - Profesión: Profesora
 - Tecnologías: C, C++, Java, HTML
 - Experiencia: Durante su formación académica como Ingeniera en telecomunicaciones, Alba ha trabajado con diferentes lenguajes de programación. En la actualidad realiza labores de formación académica a estudiantes de informática de formación profesional.
- **Juan Rey:**
 - Edad: 34
 - Profesión: Desarrollador Fullstack
 - Tecnologías: React, .NET
 - Experiencia: Juan es un desarrollador senior con más de 14 años de experiencia en el sector y en diferentes países. En la actualidad desarrolla funciones de dirección técnica en una empresa en el Reino Unido, habiéndose especializado en tecnologías web y en entornos Microsoft.
- **Orestes Carracedo:**
 - Edad: 34
 - Profesión: Desarrollador Fullstack
 - Tecnologías: Javascript, NodeJS

- **Experiencia:** Orestes es un programador Fullstack con amplia experiencia en empresas y proyectos de diferente tipología, que durante los últimos 15 años ha enfocado su carrera profesional en proyectos basados mayormente en Javascript y Cloud. Es un activo colaborador de iniciativas y formaciones orientadas a promocionar el desarrollo y las buenas prácticas.

Cuestionario general (Respuestas grupales)

<p>Te encuentras en pleno desarrollo de una funcionalidad en la aplicación. Dicha función sabes que ha sido empleada en otros proyectos de la empresa, pero no sabes en cuál.</p>	<ul style="list-style-type: none"> - Preguntar al superior jerárquico o a otros compañeros, dónde se encuentra el código a emplear - Búsqueda en internet y páginas web especializadas (p.ej. StackOverflow) - Búsqueda en los sistemas de control de código fuente (Git, TFS, etc.) - Búsqueda en sistemas de documentación (p.ej: Confluence)
<p>Acabas de escribir una función que, consideras que puede ser útil para ti en el futuro o para otros integrantes del equipo.</p>	<ul style="list-style-type: none"> - Realizar una copia de seguridad propia en algún dispositivo o cloud - Empaquetarlo en un repositorio o crear una librería. Post en la intranet de la empresa o correo electrónico informando a los integrantes del equipo - Documentar en el sistema que disponga la empresa (p.ej: Confluence)
<p>Se te ha requerido implementar una nueva función que, aunque no parece excesivamente compleja, no sabes muy bien cómo empezar.</p>	<ul style="list-style-type: none"> - Se solicita ayuda a otros integrantes del equipo - Búsqueda en internet y páginas web especializadas (p.ej. StackOverflow)
<p>En tu actividad habitual, acabas de descubrir un fragmento de código que invoca una función de utilidad cuya implementación es ineficiente o consideras que es mejorable.</p>	<ul style="list-style-type: none"> - Se realiza directamente el cambio - Se realiza directamente el cambio comentándolo con la persona responsable del equipo

	<ul style="list-style-type: none"> - Se realiza el cambio mientras se pueda verificar que la funcionalidad y sus dependencias sigan funcionando de forma correcta. En caso de existir test, éstos deberán ser ejecutados también
¿Conoces el término snippet?	<ul style="list-style-type: none"> - 50%: no - 50% sí
<u>Cuestionario específico (Respuestas grupales)</u>	
¿Qué funciones de la nueva aplicación consideras imprescindibles?	<ul style="list-style-type: none"> - Opciones y filtros de búsqueda (p.ej: por tags) - Posibilidad de crear snippets públicos y privados - Agrupación de resultados por diferentes criterios (p.ej: tags, lenguaje de programación...etc) - Búsqueda intuitiva y rápida - Multicuenta - Integración con IDE's de desarrollo - Posibilidad de crear grupos y compartir de forma privada los snippets con ellos - Ejemplos de uso
¿Qué ventajas ofrece un sistema como el descrito?	<ul style="list-style-type: none"> - Al ser un sistema web, no se depende de un sistema operativo o plataforma concreta - Copia de seguridad deslocalizada y no local - Disponer de forma organizada del conocimiento - Agilidad en el desarrollo al poder localizar de forma rápida el código que se desea emplear sin tener que buscar en internet o preguntar a otros integrantes del equipo
¿Qué desventajas le ves?	<ul style="list-style-type: none"> - Garantizar la seguridad del sistema

	<ul style="list-style-type: none"> - Olvidar que se dispone de un snippet concreto puede hacer que el sistema no sea útil
¿Empleas actualmente algún gestor de snippets? ¿Cuál? ⁹	<ul style="list-style-type: none"> - 33% (1 persona): no - 66% (2 personas): sí <p>Gestores de snippets utilizados:</p> <ul style="list-style-type: none"> - Google Chrome (cloud) - Bookmarklets (cloud) - SQL Prompt (local) - Resharper (local)
En caso de no utilizar ninguno, ¿Cuáles son las razones?	<p>Los motivos por los cuales algunos individuos (4 personas) no empleaban gestores de snippets son:</p> <ul style="list-style-type: none"> - Desconocimiento (75%) - Falta de necesidad (25%): debido a que lleva trabajando varios años en la misma empresa y proyecto, no ha surgido en ningún momento la necesidad

Tabla 9: Cuestionario general y específico DCU

3.1.1.2 Conclusiones

Tras las entrevistas grupales realizadas, se ha revelado la enorme utilidad de éstas, al poder compartir y entender a otros profesionales y sus puntos de vista.

Una parte importante de las respuestas evidencian también que los años de experiencia profesional son un factor importante a la hora de que un usuario pueda considerar la necesidad de emplear un sistema como el propuesto.

Aquellos perfiles más “junior” no llegan a valorar esta problemática, debido a que se asume que algún compañero o cualquier buscador de internet arrojará una solución al problema que se desea resolver.

⁹ Sólo aplicable para aquellos individuos que conocían el significado del término “snippet”

Sin embargo, aquellos usuarios más “senior”, y al contar con una experiencia más sólida en el ámbito del desarrollo software, reconocen la utilidad de contar con un sistema como el descrito, en aras de la productividad y agilidad en las actividades diarias.

Se recalca además la posibilidad más que factible, de que uno no trabaje siempre en la misma empresa, con lo cual el acceso a ciertos fragmentos de código del pasado, ya no se resultaría posible.

Por último, dichas entrevistas han demostrado también su utilidad para descubrir nuevas funciones potenciales, que darían más valor al producto.

Aquellas más reseñables pero que por desgracia se quedarían - por una cuestión de planificación y tiempo - fuera del alcance del proyecto son:

- Creación de grupos de usuarios para compartir fragmentos de código de forma privada (p.ej: un equipo de trabajo de una empresa)
- Integración mediante una API ad-hoc con componentes de terceros
- Desarrollo de plugins para otros entornos de desarrollo o navegadores (Chrome, VSCode, IntelliJ, etc.)

3.1.2 Flujos de interacción

Un diagrama de flujo¹⁰ es un gráfico representativo de un proceso, o la solución esquemática de un problema mediante figuras conectadas por líneas, con el fin de diseñar o documentar un proceso o programa.

Siguiendo este principio, se ha realizado un diagrama que representa la solución a implementar con las distintas actividades o acciones que pueden llevarse a cabo en el mismo.

Se distingue además mediante la agrupación pertinente entre aquellas actividades que pueden ser llevadas a cabo por usuarios anónimos y aquellas que deben ser llevadas a cabo por usuarios autenticados en el sistema.

De la misma forma, para aquellos usuarios ya autenticados se diferencia entre aquellas acciones que sólo pueden ser llevadas a cabo por un administrador, y aquellas que no requieren dicho rol. Por último, y con el

¹⁰ La definición formal según la ISO/IEC 2382:2015 puede encontrarse en el siguiente enlace: <https://www.iso.org/obp/ui/- iso:std:iso-iec:2382:ed-1:v1:en>

fin de aportar mayor claridad al diagrama, existen determinadas acciones que estarán disponibles o no en virtud de si el usuario es creador o no de un determinado snippet.

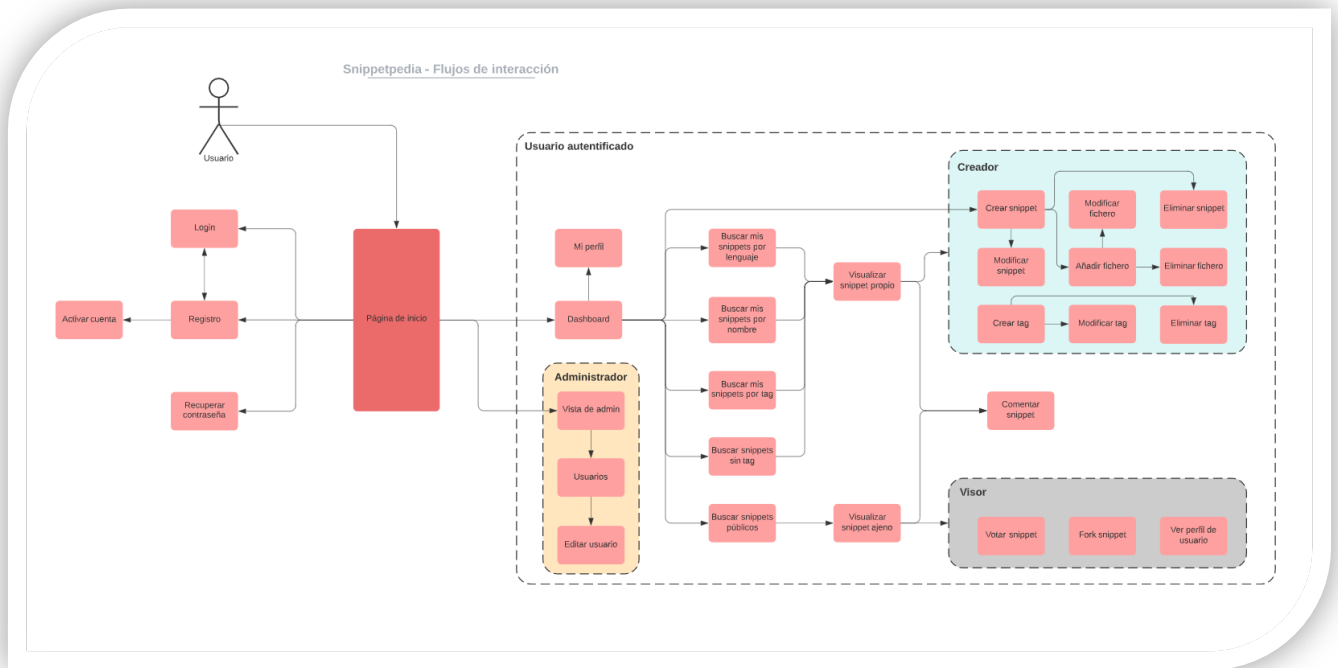


Figura 6: Diagrama de flujos de interacción

3.1.3 Prototipado

En el presente capítulo se presentan varios prototipos¹¹ referidos a las funciones anteriormente visuales, así como una breve descripción contextual y funcional de los mismos.

3.1.3.1 Página de inicio

Prototipo	Descripción
	<p>Al acceder al sitio web principal, se presentará la clásica información de un producto en forma de <i>“Landing Page”</i>.</p> <p>Dicha información incorporará detalles tales como capturas de pantalla, enumeración de las</p>

¹¹ Los prototipos del presente documento han sido elaborados con la aplicación Moqups. Más información: <http://www.moqups.com/>

	<p>funcionalidades principales, enlaces a redes sociales y un acceso rápido a las dos funciones más habituales:</p> <ul style="list-style-type: none"> - Registro de un nuevo usuario (3.1.3.2) - Acceso al dashboard para los usuarios ya existentes (3.1.3.3)
--	---

Tabla 10: Prototipo página de inicio

3.1.3.2 Registro de usuarios

Prototipo	Descripción
	<p>A través de esta funcionalidad, un usuario podrá darse de alta en el sistema introduciendo una serie de datos básicos, así como sus credenciales en último lugar.</p> <p>En la misma pantalla existirá un enlace que muestre los términos de uso y política de privacidad.</p> <p>En caso de que un usuario hubiese accedido por error a la página de registro y se haya percatado de</p>


	<p>que su usuario ya existe, para mayor comodidad se le ofrecerá la posibilidad de ser redireccionado a la página de login (3.1.3.3).</p> <p>Tras la creación exitosa del usuario, éste deberá recibir en su correo electrónico un enlace de activación que permitirá verificar la autenticidad de la cuenta de correo electrónico introducida.</p>
---	---

Tabla 11: Prototipo registro de usuario

3.1.3.3 Login


Prototipo	Descripción
	<p>Tras la creación exitosa de un usuario, éste podrá acceder a las funciones básicas del sistema tras autenticarse en el mismo.</p> <p>Para ello, y mediante el enlace dispuesto en la página de inicio, éste deberá introducir los datos de acceso de su registro previo de usuario.</p> <p>En caso de que éste no los recuerde, se ofrece la posibilidad de reestablecer la contraseña mediante el enlace creado para ello (3.1.3.4).</p> <p>De forma análoga al registro de usuario, en caso de que éste se percate de que su cuenta no existe, se dispone para mayor comodidad de un enlace directo al registro de nuevos usuarios.</p>

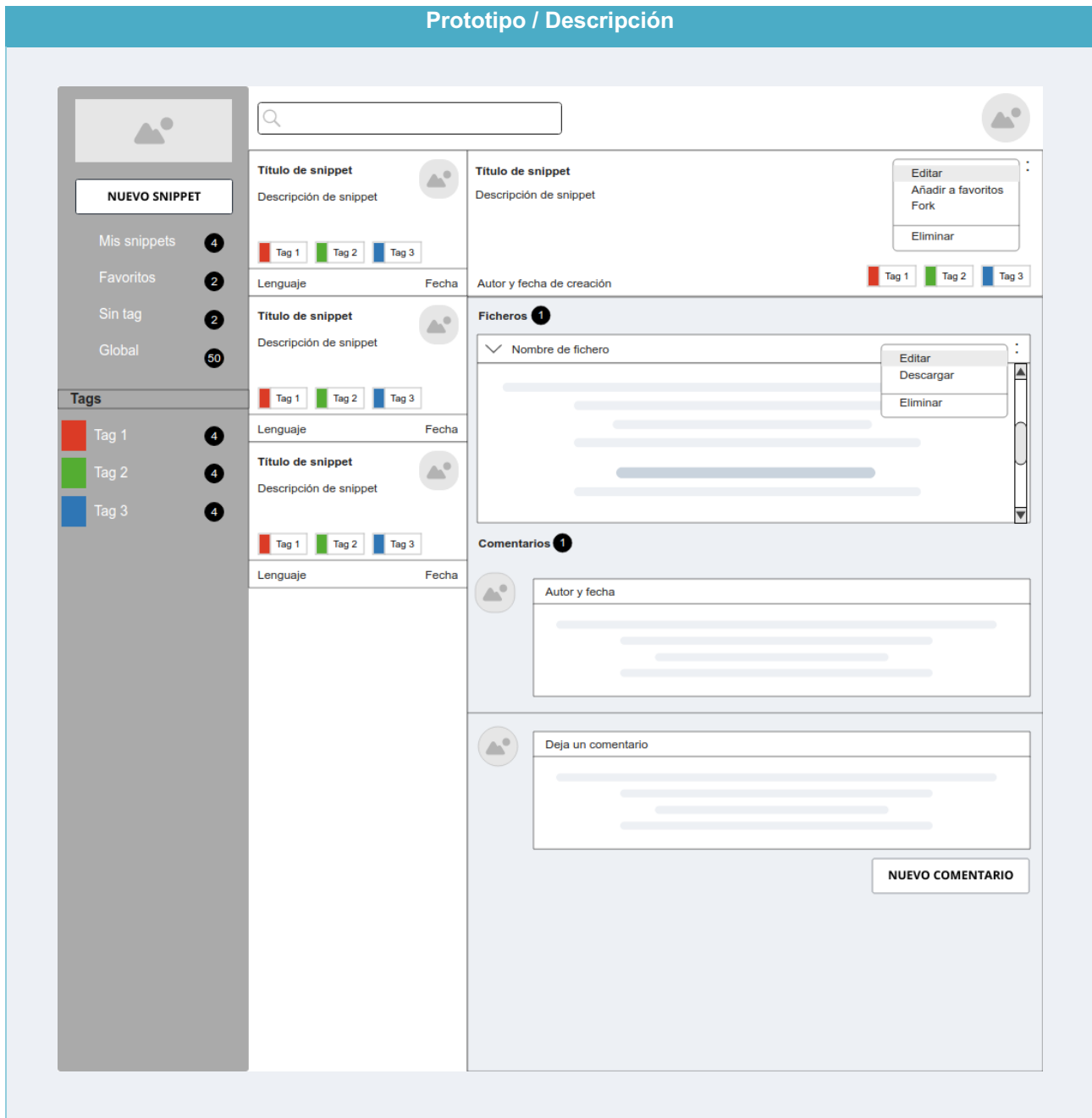
Tabla 12: Prototipo login de usuario

3.1.3.4 Reestablecer contraseña

Prototipo	Descripción
 <p>El prototipo muestra una interfaz de usuario para reestablecer la contraseña. En la parte superior, hay un icono de perfil de usuario. El formulario principal tiene el título "Reestablecer contraseña". Debajo del título, hay un campo de entrada etiquetado "Correo electrónico". Debajo del campo de entrada, hay un botón "ACCEDER". En la parte inferior del formulario, hay un texto descriptivo "Texto descriptivo creación cuenta" y un botón "REGISTRO".</p>	<p>El restablecimiento de la contraseña permite a un usuario que no recuerde sus datos de acceso acceder a un enlace especial recibido por correo electrónico, que le permitirá obtener una nueva contraseña.</p>

Tabla 13: Prototipo reestablecer contraseña

3.1.3.5 Dashboard



El dashboard es el componente principal del sistema y el más complejo a nivel de funcionalidades y usabilidad. El objetivo principal es que el usuario pueda realizar las acciones deseadas de una forma ágil, simple, intuitiva y con la menor cantidad de “clicks” posibles.

Por eso se propone una disposición de la página dividida en tres columnas:

- **Menú de navegación:** este menú ofrecerá un acceso rápido a la creación de nuevos snippets, así como búsquedas pre-filtradas según diferentes criterios: snippets propios, favoritos, sin tag o con tags específicos.
- **Columna de snippets y resultados:** tras acceder a algún elemento del menú de navegación, se cargará en esta segunda columna aquellos snippets que cumplan los criterios de búsqueda especificados. Por cada snippet se mostrarán datos tales como el nombre, descripción, tags y un acceso a la ficha del usuario propietario del mismo.
- **Detalle de snippet:** tras acceder a algún snippet concreto de la segunda columna, se mostrará en la tercera columna los detalles extendidos del mismo.
 - o En la parte superior de la tercera columna se localizarán los datos primarios del snippet. También se mostrarán en forma de menú contextual o de barra de botones, accesos directos a funciones tales como modificar, eliminar o añadir un nuevo fichero.
 - o A continuación, se enumerarán los diferentes ficheros que componen el snippet. Entre otras funciones se podrá por cada fichero modificar o eliminar el mismo.
 - o En la parte inferior se mostrarán la lista de comentarios de los diferentes usuarios ordenados de forma cronológica, localizándose los comentarios más recientes en la parte superior. Por cada comentario – y mientras el usuario actual sea el autor de éste – se mostrarán opciones adicionales para modificar o eliminar un comentario.
- **Búsqueda de snippet:** En la parte superior de la página se mostrará una caja de texto que permitirá realizar un filtrado adicional sobre la lista de snippets mostrados en la segunda columna. Dicha búsqueda afectará a los campos nombre y descripción del snippet.
- **Perfil de usuario:** En la parte derecha superior se podrá acceder mediante menú contextual a la ficha de perfil del usuario actual o un enlace para finalizar la sesión actual.

Tabla 14: Prototipo dashboard

3.1.3.6 Crear / modificar snippet

Prototipo	Descripción
	<p>Mediante el botón disponible en el menú lateral del dashboard (3.1.3.5), se mostrará un formulario modal que permitirá crear un nuevo snippet con los datos básicos del mismo, los cuales son:</p> <ul style="list-style-type: none"> - Nombre - Descripción

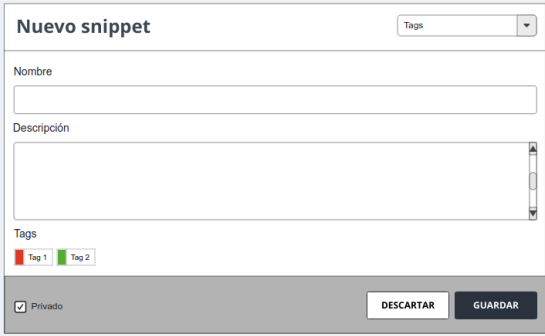
	<p>- Tags</p> <p>La selección de tags (3.1.3.7) se realizará mediante un componente creado específicamente para dicho fin, que permitirá seleccionar todos aquellos tags que son relevantes para el snippet a crear. Tras dicha selección, éstos serán mostrados en la parte inferior.</p>
---	--

Tabla 15: Prototipo creación y modificación de snippet

3.1.3.7 Seleccionar tags

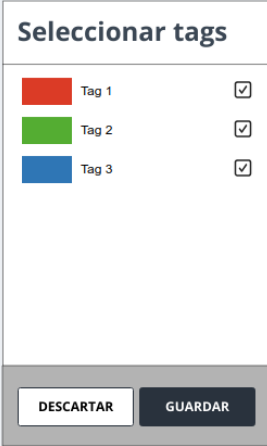
Prototipo	Descripción
	<p>La selección de tags es un componente que permite selección múltiple de tags con el fin de que el usuario no tenga que escribirlos individualmente.</p> <p>Dicho componente será empleado en el contexto de la creación, edición y búsqueda de snippets.</p>

Tabla 16: Prototipo seleccionar tags

3.1.3.8 Crear / modificar fichero

Prototipo	Descripción
	<p>El formulario para crear o editar un fichero puede ser mostrado tras la creación de un snippet. Como datos básicos del mismo deberá establecerse un</p>

Nuevo fichero	
Nombre	Tipo de fichero
<input type="text"/>	C#
Contenido	
<div style="border: 1px solid #ccc; height: 100px; padding: 5px;">[Placeholder for content]</div>	
<input type="button" value="DESCARTAR"/> <input type="button" value="GUARDAR"/>	


nombre y un lenguaje de programación o tipo de fichero.

Por último, se deberá introducir el contenido del fichero en el campo de texto denominado como "Contenido", el cual aplicará un resaltado u otro según la sintaxis del lenguaje de programación seleccionado.

Tabla 17: Prototipo creación y modificación de fichero

3.1.3.9 Perfil de usuario

Prototipo / Descripción



Alias Usuario
Localidad, País

40 Snippets | 10 Tags

Bio

[Placeholder for bio]

Tags

[Placeholder for tags]

Últimos comentarios

[Placeholder for comments]

Mediante el formulario de visualización de un perfil de usuario, se podrán consultar informaciones básicas públicas del mismo, como la biografía y estadísticas varias entre otros.

Tabla 18: Prototipo perfil de usuario

3.2 Diseño técnico

La información recopilada en las diferentes entrevistas y el análisis de las aplicaciones existentes establecen una sólida base en la cual se fundamentará la definición funcional y técnica del proyecto.

3.2.1 Actores del sistema

En el presente sistema se diferencian dos posibles actores:

- Usuario no registrado: aquel que todavía no ha finalizado el proceso de registro y que, por ende, no puede acceder al sistema.
- Usuario registrado: aquel que ya ha finalizado el proceso de registro y cuenta con unos datos de acceso válidos.

3.2.2 Diagramas de casos de uso

Los diagramas de casos de uso son una técnica de la ingeniería del software para la captura de requisitos potenciales de un sistema. La definición de los casos de uso nos presenta los escenarios que definen cómo debe ser la interacción de los denominados actores con uno o varios sistemas para conseguir un objetivo determinado.

3.2.2.1 Diagrama de paquetes de casos de uso

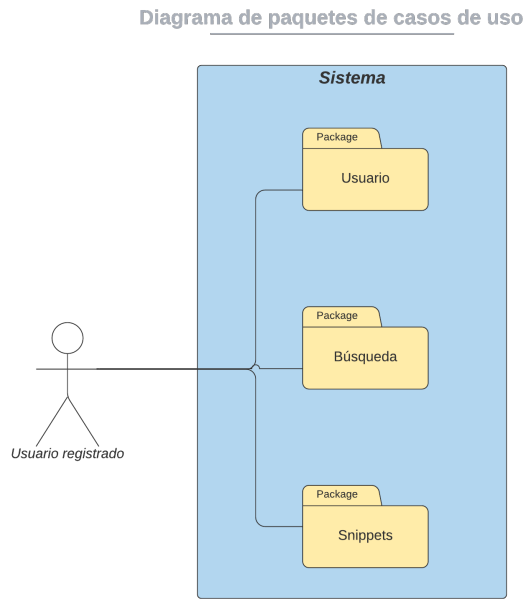


Figura 7: Diagrama de paquetes de casos de uso

3.2.2.2 Diagrama de casos de uso: Usuarios

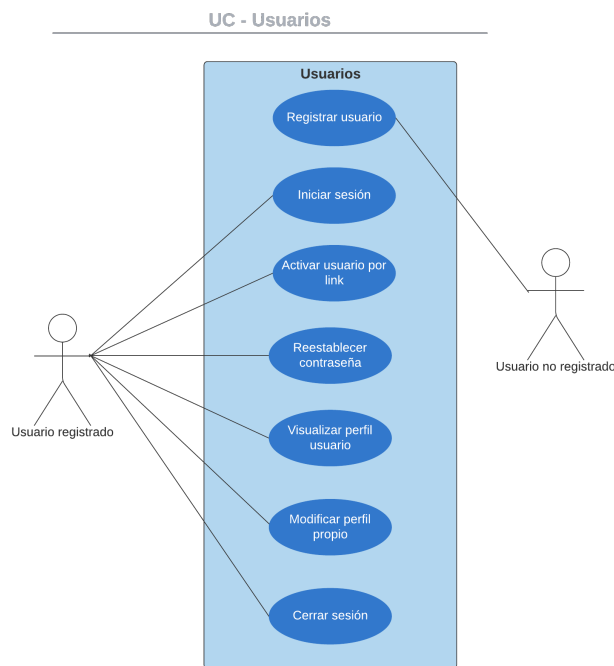


Figura 8: Diagrama de casos de uso: Usuarios

3.2.2.3 Diagrama de casos de uso: Búsquedas

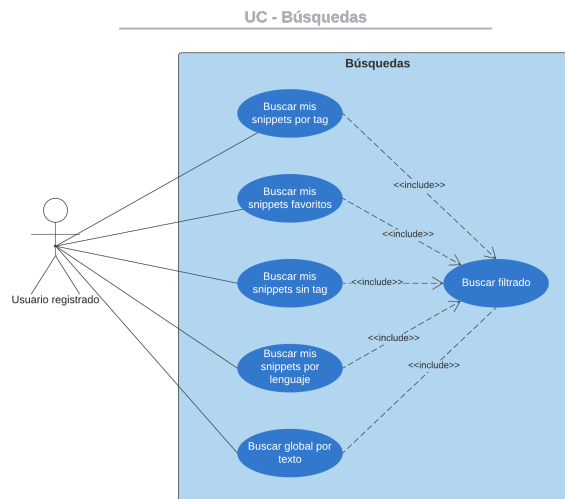


Figura 9: Diagrama de casos de uso: Búsquedas

3.2.2.4 Diagrama de casos de uso: Snippets

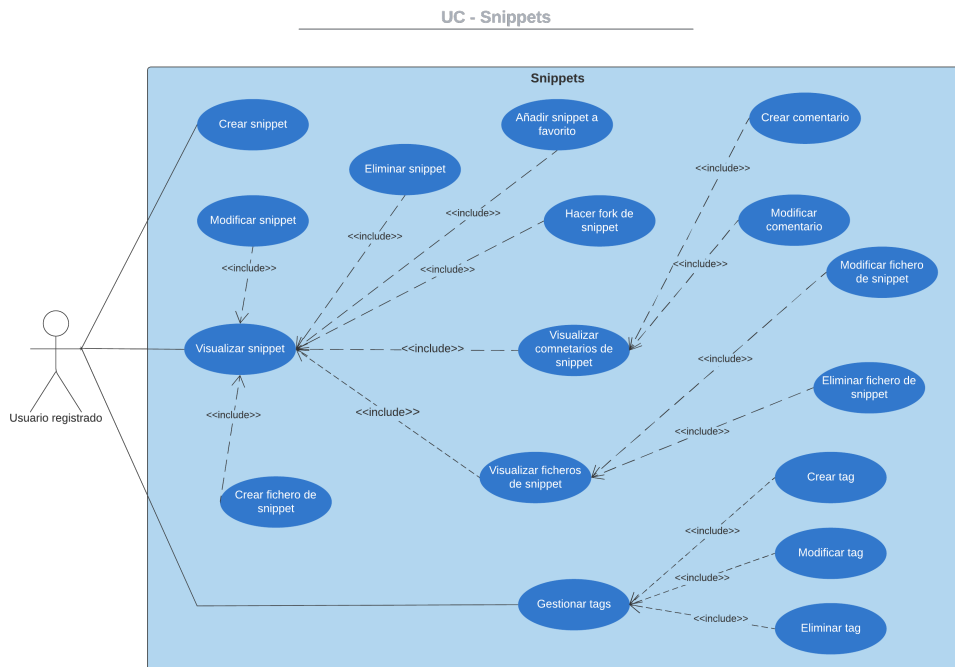


Figura 10: Diagrama de casos de uso: Snippets

3.2.3 Listado de casos de uso

CU001 Registrar usuario	
Descripción	Permite al usuario crear una cuenta en el sistema para acceder a las funcionalidades
Actores	Usuario no registrado
Precondiciones	<ul style="list-style-type: none"> - El correo electrónico no debe existir en el sistema
Flujo	<ul style="list-style-type: none"> - Acceder al formulario de registro - Cumplimentar aquellos campos marcados como obligatorios - Pulsar el botón que permite guardar los cambios
Excepciones	<ul style="list-style-type: none"> - Los campos obligatorios no han sido cumplimentados - El correo electrónico ya existe - El alias ya existe - La contraseña no cumple con los parámetros de complejidad necesarios
Postcondiciones	<ul style="list-style-type: none"> - La cuenta de usuario será creada - El usuario recibirá un correo electrónico con un enlace para activar su cuenta
Comentarios	-

CU002 Iniciar sesión	
Descripción	Permite a un usuario registrado acceder al sistema
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha sido registrado (CU001) - Recordar datos de acceso - El usuario ha sido activado
Flujo	<ul style="list-style-type: none"> - Acceder al formulario de inicio de sesión - Introducir datos de acceso - Pulsar el botón de acceso
Excepciones	<ul style="list-style-type: none"> - Los campos obligatorios no han sido cumplimentados - Los datos de acceso no son correctos - El usuario no ha sido activado (CU003)
Postcondiciones	<ul style="list-style-type: none"> - El usuario será redirigido a su dashboard
Comentarios	-

CU003 Activar usuario por link	
Descripción	Activa a un usuario existente

Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha sido registrado (CU001) - El usuario ha recibido el correo con el link de activación - El usuario no ha sido activado
Flujo	<ul style="list-style-type: none"> - Abrir el correo electrónico recibido - Pulsar en el enlace de activación
Excepciones	<ul style="list-style-type: none"> - El enlace de activación no es correcto - El usuario ya ha sido previamente activado
Postcondiciones	<ul style="list-style-type: none"> - El usuario será redirigido a una página informativa con el resultado de la activación
Comentarios	-

CU004 Reestablecer contraseña	
Descripción	El usuario podrá, previa confirmación por correo electrónico, reestablecer su contraseña
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha sido registrado (CU001) - El usuario ha sido activado (CU003) - El usuario no recuerda sus datos de acceso
Flujo	<ul style="list-style-type: none"> - Acceder al enlace de restablecimiento de contraseña - Introducir correo electrónico - Pulsa botón de confirmación - Recibir correo electrónico con enlace específico - Pulsa en enlace específico - Introducir la nueva contraseña - Pulsar el botón que permite guardar los cambios
Excepciones	<ul style="list-style-type: none"> - El usuario no existe - El usuario no ha sido activado
Postcondiciones	<ul style="list-style-type: none"> - El usuario será redirigido a una página en la cual podrá introducir su nueva contraseña - El usuario confirmará la operación pulsando el botón adecuado
Comentarios	-

CU005 Visualizar perfil de usuario	
Descripción	El usuario podrá en cualquier momento y mientras la imagen y el alias de un usuario sea visible, acceder al perfil de un usuario concreto

Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - En algún componente se muestra un enlace con el alias o la foto de un usuario de la aplicación
Flujo	<ul style="list-style-type: none"> - En el uso habitual de la aplicación se muestra un enlace con el alias o la imagen de un usuario de la aplicación - El usuario pulsa sobre el enlace o imagen de un usuario de la aplicación
Excepciones	-
Postcondiciones	- El usuario será redirigido a la página de visualización de perfil de un usuario
Comentarios	-

CU006 Modificar perfil propio	
Descripción	El usuario puede modificar sus datos de perfil con los que se registró en su momento
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002)
Flujo	<ul style="list-style-type: none"> - El usuario accede pulsando en su propia imagen o avatar a un menú que le permite acceder a su perfil - El usuario modificar cualquier dato de su perfil a excepción del correo electrónico - Pulsar el botón que permite guardar los cambios
Excepciones	-
Postcondiciones	- Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU007 Cerrar sesión	
Descripción	El usuario finaliza su sesión actual
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002)
Flujo	<ul style="list-style-type: none"> - El usuario accede pulsando en su propia imagen o avatar a un menú que le permite finalizar su sesión actual
Excepciones	-
Postcondiciones	- El usuario será redirigido a una página en la que se le informa de que su sesión ha sido finalizada

Comentarios	-
--------------------	---

CU008 Buscar mis snippets por tag	
Descripción	Permite localizar aquellos snippets propios que hayan sido marcados con la etiqueta correspondiente
Actores	Usuario registrado
Precondiciones	- El usuario ha iniciado sesión (CU002)
Flujo	- En el dashboard de usuario, éste accederá en el menú lateral al elemento que represente el tag o etiqueta concreta por el cual desea realizar el filtrado
Excepciones	-
Postcondiciones	- La lista de elementos coincidentes será mostrada en el contenedor adecuado - En caso de no haber elementos coincidentes, se mostrará un panel o alerta informativa
Comentarios	-

CU009 Buscar mis snippets favoritos	
Descripción	Permite localizar aquellos snippets propios y ajenos que hayan sido marcados como favoritos
Actores	Usuario registrado
Precondiciones	- El usuario ha iniciado sesión (CU002)
Flujo	- En el dashboard de usuario, éste accederá en el menú lateral al elemento que represente los favoritos
Excepciones	-
Postcondiciones	- La lista de elementos coincidentes será mostrada en el contenedor adecuado - En caso de no haber elementos coincidentes, se mostrará un panel o alerta informativa
Comentarios	-

CU010 Buscar mis snippets sin tag	
Descripción	Permite localizar aquellos snippets propios que no tengan etiqueta
Actores	Usuario registrado
Precondiciones	- El usuario ha iniciado sesión (CU002)

Flujo	- En el dashboard de usuario, éste accederá en el menú lateral al elemento que represente los snippets sin etiqueta
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - La lista de elementos coincidentes será mostrada en el contenedor adecuado - En caso de no haber elementos coincidentes, se mostrará un panel o alerta informativa
Comentarios	-

CU011 Buscar mis snippets por lenguaje

Descripción	Permite localizar aquellos snippets propios que contengan uno o más ficheros de un determinado lenguaje de programación
Actores	Usuario registrado
Precondiciones	- El usuario ha iniciado sesión (CU002)
Flujo	- En el dashboard de usuario, éste accederá en el menú lateral al elemento que represente los snippets de un determinado lenguaje de programación
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - La lista de elementos coincidentes será mostrada en el contenedor adecuado - En caso de no haber elementos coincidentes, se mostrará un panel o alerta informativa
Comentarios	-

CU012 Buscar global por texto

Descripción	Permite localizar aquellos snippets ajenos que coincidan con el texto introducido en la barra de búsqueda
Actores	Usuario registrado
Precondiciones	- El usuario ha iniciado sesión (CU002)
Flujo	<ul style="list-style-type: none"> - El usuario escribirá un texto en la barra de búsqueda superior - El usuario accederá en el menú lateral sobre el elemento búsqueda global
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - La lista de elementos coincidentes será mostrada en el contenedor adecuado - En caso de no haber elementos coincidentes, se mostrará un panel o alerta informativa
Comentarios	-

CU013 Crear snippet	
Descripción	Crea un nuevo snippet propio
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002)
Flujo	<ul style="list-style-type: none"> - Pulsar a través del menú lateral en el botón que permite crear un nuevo snippet - Introducir los datos obligatorios - Seleccionar si es preciso los tags o etiquetas a emplear - Pulsar el botón que permite guardar los cambios
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - Se creará un nuevo snippet asociado al usuario - Se preseleccionará el nuevo snippet en el menú lateral mostrando la información adicional del mismo - Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU014 Modificar snippet	
Descripción	Modifica un snippet propio ya existente
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario ha creado un snippet (CU013)
Flujo	<ul style="list-style-type: none"> - Seleccionar un snippet ya existente - Se mostrará un panel de detalle del snippet - Pulsar en el botón modificar - Se mostrará una ventana con los datos ya almacenados - Realizar las modificaciones oportunas - Pulsar el botón que permite guardar los cambios
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - Al guardar los cambios se refrescará la información mostrada y se preseleccionará de nuevo el snippet modificado - Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU015 Eliminar snippet	
Descripción	Elimina un snippet propio, así como los ficheros asociados al mismo
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario ha creado un snippet (CU013)
Flujo	<ul style="list-style-type: none"> - Seleccionar un snippet ya existente - Se mostrará un panel de detalle del snippet - Pulsar en el botón eliminar
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - El snippet y todas las informaciones asociadas serán eliminadas - Los usuarios que tuviesen el snippet en sus favoritos no lo podrán visualizar más - Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU016 Visualizar snippet	
Descripción	Muestra la información de un snippet
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario creador siempre puede ver todos sus snippets, sean privados o públicos - Sólo aquellos snippets públicos podrán ser accedidos por otros usuarios
Flujo	<ul style="list-style-type: none"> - Realizar una búsqueda de un snippet, independientemente del criterio (CU008 CU009 CU010 CU011 CU012) - Pulsar sobre el snippet deseado
Excepciones	- Sólo los snippets públicos pueden ser visualizados por terceros usuarios
Postcondiciones	- Al acceder al elemento del contenedor de snippets se cargará la información asociada al mismo
Comentarios	-

CU017 Crear tag	
Descripción	Crea una etiqueta personal asociada al usuario actual
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002)

Flujo	<ul style="list-style-type: none"> - Pulsar el botón de gestión de tags - Pulsar el botón de creación de tags - Complimentar aquellos campos marcados como obligatorios - Pulsar el botón que permite guardar los cambios
Excepciones	<ul style="list-style-type: none"> - No es posible crear dos tags con el mismo nombre - No es posible crear dos tags con el mismo color
Postcondiciones	<ul style="list-style-type: none"> - Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU018 Modificar tag	
Descripción	Modifica una etiqueta personal
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario ha creado un tag (CU017)
Flujo	<ul style="list-style-type: none"> - Pulsar el botón de gestión de tags - Pulsar sobre el tag que se desee modificar - Introducir los datos obligatorios - Pulsar el botón que permite guardar los cambios
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU019 Eliminar tag	
Descripción	Elimina una etiqueta personal
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario ha creado un tag (CU017)
Flujo	<ul style="list-style-type: none"> - Pulsar el botón de gestión de tags - Pulsar sobre el tag que se desee eliminar
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - Se eliminará el tag de aquellos snippets que referenciasen el tag seleccionado - Se muestra un panel informativo indicando el éxito o fracaso de la operación

Comentarios	-
--------------------	---

CU020 Visualizar comentarios de snippet	
Descripción	Muestra los comentarios de un determinado snippet
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario ha creado un snippet (CU013)
Flujo	<ul style="list-style-type: none"> - Visualizar un snippet (CU016)
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - Se mostrarán de forma cronológica los comentarios creados por los usuarios
Comentarios	-

CU021 Visualizar ficheros de snippet	
Descripción	Muestra los ficheros de un determinado snippet
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario ha creado un snippet (CU013)
Flujo	<ul style="list-style-type: none"> - El usuario ha buscado y visualizado un snippet (CU016)
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - Se mostrarán de forma alfabética los ficheros añadidos por el usuario creador
Comentarios	-

CU022 Crear comentario	
Descripción	Crea un comentario para un determinado snippet
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario ha creado un snippet (CU013) - El usuario ha buscado y visualizado un snippet (CU016)
Flujo	<ul style="list-style-type: none"> - Pulsar el botón para crear un comentario - Introducir los datos obligatorios - Pulsar el botón que permite guardar los cambios
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - Se muestra un panel informativo indicando el éxito o fracaso de la operación

Comentarios	-
--------------------	---

CU023 Modificar comentario	
Descripción	Modifica un comentario para un determinado snippet
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - Un usuario ha creado un snippet (CU013) - El usuario ha buscado y visualizado un snippet (CU016) - Visualizar los comentarios (CU020)
Flujo	<ul style="list-style-type: none"> - Localizar el comentario a modificar - Pulsar el botón para modificar el comentario - Introducir los datos obligatorios - Pulsar el botón que permite guardar los cambios
Excepciones	<ul style="list-style-type: none"> - Sólo es posible modificar los comentarios propios
Postcondiciones	<ul style="list-style-type: none"> - Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU024 Crear fichero de snippet	
Descripción	Crea un nuevo fichero en un snippet propio
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - Un usuario ha creado un snippet (CU013) - El usuario ha buscado y visualizado un snippet (CU016)
Flujo	<ul style="list-style-type: none"> - Pulsar el botón para crear un fichero - Introducir los datos obligatorios - Pulsar el botón que permite guardar los cambios
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU025 Modificar fichero de snippet	
Descripción	Modifica un fichero existente de un snippet propio
Actores	Usuario registrado

Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - Un usuario ha creado un snippet (CU013) - El usuario ha buscado y visualizado un snippet (CU016) - Visualizar los ficheros (CU021)
Flujo	<ul style="list-style-type: none"> - Localizar el fichero a modificar - Pulsar el botón para modificar el fichero - Introducir los datos obligatorios - Pulsar el botón que permite guardar los cambios
Excepciones	<ul style="list-style-type: none"> - Sólo el autor del snippet puede modificar los ficheros
Postcondiciones	<ul style="list-style-type: none"> - Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU026 Eliminar fichero de snippet	
Descripción	Elimina un fichero existente de un snippet propio
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - Un usuario ha creado un snippet (CU013) - El usuario ha buscado y visualizado un snippet (CU016) - Visualizar los ficheros (CU021)
Flujo	<ul style="list-style-type: none"> - Localizar el fichero a eliminar - Pulsar el botón para eliminar el fichero - Pulsar el botón que permite guardar los cambios
Excepciones	-
Postcondiciones	<ul style="list-style-type: none"> - Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU027 Añadir snippet a favorito	
Descripción	Añade un snippet a la lista de favoritos
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario ha creado un snippet (CU013) - El usuario ha buscado y visualizado un snippet (CU016)
Flujo	<ul style="list-style-type: none"> - Pulsar el botón para añadir un snippet a la lista de favoritos
Excepciones	-

Postcondiciones	- Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

CU028	Hacer fork de snippet
Descripción	Hace una copia de un snippet ajeno como propio
Actores	Usuario registrado
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión (CU002) - El usuario ha creado un snippet (CU013) - El usuario ha buscado y visualizado un snippet (CU016)
Flujo	- Pulsar el botón para crear un fork de un snippet público
Excepciones	- Sólo se pueden realizar forks de snippets de otros usuarios
Postcondiciones	- Se muestra un panel informativo indicando el éxito o fracaso de la operación
Comentarios	-

3.2.4 Arquitectura

Como se ha descrito en capítulos anteriores, el sistema a desarrollar se tratará de una aplicación Web basada en React. Dicha tecnología permite el desarrollo de páginas web SPA (Single Page Application) en la cual los recursos estáticos necesarios son cargados una sola vez de forma que el usuario tenga la percepción de que siempre se encuentra en la misma página, al no existir transiciones.

Aunque para el desarrollo de la aplicación web se utilizarán técnicas y frameworks *responsive*¹², debido a la naturaleza y el uso que se le va a dar, se estima que en la mayor parte de los casos será empleada desde un ordenador personal.

La aplicación web deberá obtener los datos y realizar peticiones contra una API Rest expuesta con tal fin desarrollada con .NET Core. El sistema backend se dividirá en diferentes capas (API, negocio, persistencia) y se basará en frameworks maduros y consolidados en materia de seguridad y validación de los datos con el fin de garantizar la robustez de este. Los datos serán en último lugar almacenados en una base de datos SQL Server.

El sistema en conjunto será alojado en Azure y mediante el uso de CI/CD como se explica en el capítulo 3.2.5, será posible automatizar los despliegues en el entorno de producción.

¹² Técnica de diseño web que permite la correcta visualización de una página en distintos dispositivos

3.2.4.1 Visión general

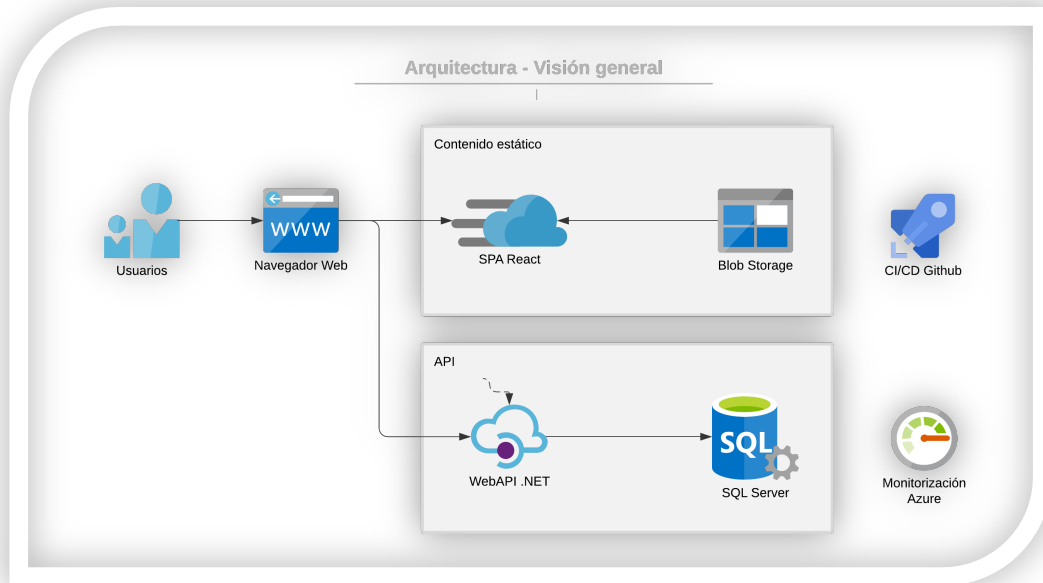


Figura 11: Arquitectura - Visión general

3.2.4.2 Modelo de datos

A continuación, se describe de forma diagramática y tabular la estructura de la base de datos del sistema a desarrollar.

3.2.4.2.1 Diagrama

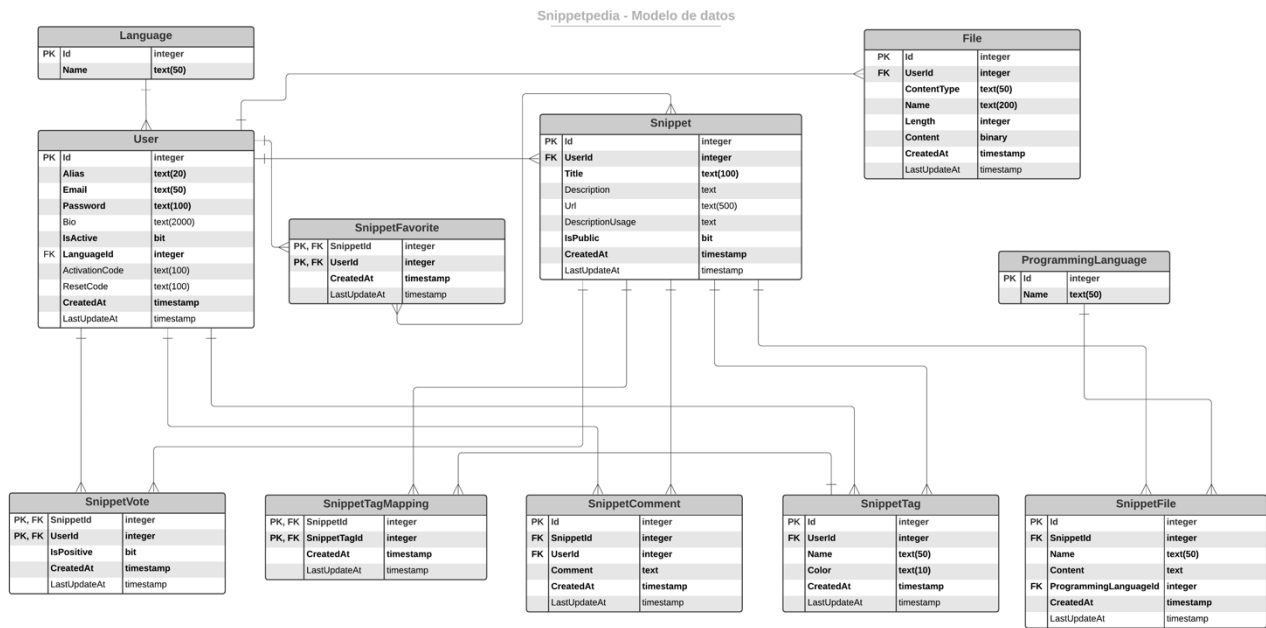


Figura 12: Diagrama de base de datos

3.2.4.2.2 Tablas

En el presente capítulo se describen de una forma más detallada las tablas de la base de datos del sistema a desarrollar.

User			
Descripción	Usuarios de la aplicación y sus preferencias		
Atributo	Descripción	Tipo	Nullable
Id	Clave primaria, columna incremental	Integer	No
Alias	Alias público del usuario	Text(20)	No
Email	Correo electrónico	Text(50)	No
Password	Contraseña cifrada	Text(100)	No
Bio	Información pública del usuario	Text(2000)	Sí
IsActive	Especifica si un usuario ha activado su cuenta	Bit	No
LanguageId	Idioma	Integer	No
ActivationCode	Código de activación	Text(100)	Sí

ResetCode	Código de restablecimiento de contraseña	Text(100)	Sí
CreatedAt	Fecha de creación	Timestamp	No
LastUpdateAt	Fecha de la última actualización	Timestamp	Sí
Claves primarias	Id		
Claves foráneas	LanguageId		

File			
Descripción	Tabla de ficheros de usuario (p.ej: avatares)		
Atributo	Descripción	Tipo	Nullable
Id	Clave primaria, columna incremental	Integer	No
UserId	Id de usuario	Integer	No
ContentType	Tipo de contenido	Text(50)	No
Length	Tamaño del fichero en bytes	Integer	No
Content	Contenido en binario del fichero	Binary	No
CreatedAt	Fecha de creación	Timestamp	No
LastUpdateAt	Fecha de la última actualización	Timestamp	Sí
Claves primarias	Id		
Claves foráneas	UserId		

Language			
Descripción	Idiomas seleccionables de la aplicación		
Atributo	Descripción	Tipo	Nullable
Id	Clave primaria, columna incremental	Integer	No
Name	Nombre	Text(50)	No
Claves primarias	Id		
Claves foráneas	-		

ProgrammingLanguage	
Descripción	Lenguajes de programación asociables a un fichero de un snippet

Atributo	Descripción	Tipo	Nullable
Id	Clave primaria, columna incremental	Integer	No
Name	Nombre	Text(50)	No
Claves primarias	Id		
Claves foráneas	-		

Snippet			
Descripción	Fragmentos de código de un usuario		
Atributo	Descripción	Tipo	Nullable
Id	Clave primaria, columna incremental	Integer	No
UserId	Id de usuario	Integer	No
Title	Título	Text(100)	No
Description	Descripción	Text	No
Url	Enlace web de referencia	Text(500)	Sí
DescriptionUsage	Texto descriptivo de uso del snippet	Text	Sí
IsPublic	Público / Privado	Bit	No
CreatedAt	Fecha de creación	Timestamp	No
LastUpdateAt	Fecha de la última actualización	Timestamp	Sí
Claves primarias	Id		
Claves foráneas	UserId		

SnippetTag			
Descripción	Etiquetas personales de un usuario para su lista de fragmentos de código		
Atributo	Descripción	Tipo	Nullable
Id	Clave primaria, columna incremental	Integer	No
UserId	Id de usuario	Integer	No
Name	Nombre descriptivo	Text(50)	No
Color	Color	Text(10)	No
CreatedAt	Fecha de creación	Timestamp	No
LastUpdateAt	Fecha de la última actualización	Timestamp	Sí
Claves primarias	Id		

Claves foráneas	UserId
------------------------	--------

SnippetTagMapping			
Descripción	Lista de etiquetas de un snippet		
Atributo	Descripción	Tipo	Nullable
SnippetId	Id del snippet	Integer	No
SnippetTagId	Id de la etiqueta	Integer	No
CreatedAt	Fecha de creación	Timestamp	No
LastUpdateAt	Fecha de la última actualización	Timestamp	Sí
Claves primarias	SnippetId, SnippetTagId		
Claves foráneas	SnippetId, SnippetTagId		

SnippetFile			
Descripción	Ficheros de un snippet		
Atributo	Descripción	Tipo	Nullable
Id	Clave primaria, columna incremental	Integer	No
SnippetId	Id del snippet	Integer	No
Name	Nombre descriptivo	Text(50)	No
Content	Contenido textual del snippet	Text	No
ProgrammingLanguageId	Id del lenguaje de programación	Integer	No
CreatedAt	Fecha de creación	Timestamp	No
LastUpdateAt	Fecha de la última actualización	Timestamp	Sí
Claves primarias	Id		
Claves foráneas	SnippetId, ProgrammingLanguageId		

SnippetFavorite			
Descripción	Fragmentos de código favoritos de un usuario		
Atributo	Descripción	Tipo	Nullable
SnippetId	Id del snippet	Integer	No
UserId	Id del usuario	Integer	No

CreatedAt	Fecha de creación	Timestamp	No
LastUpdatedAt	Fecha de la última actualización	Timestamp	Sí
Claves primarias	SnippetId, UserId		
Claves foráneas	SnippetId, UserId		

SnippetVote

Descripción	Votos a un fragmento de código por parte de los usuarios		
Atributo	Descripción	Tipo	Nullable
SnippetId	Id del snippet	Integer	No
UserId	Id del usuario	Integer	No
IsPositive	Positivo / Negativo	Bit	No
CreatedAt	Fecha de creación	Timestamp	No
LastUpdatedAt	Fecha de la última actualización	Timestamp	Sí
Claves primarias	SnippetId, UserId		
Claves foráneas	SnippetId, UserId		

SnippetComment

Descripción	Comentarios de los usuarios acerca de un fragmento de código		
Atributo	Descripción	Tipo	Nullable
Id	Clave primaria, columna incremental	Integer	No
SnippetId	Id del snippet	Integer	No
UserId	Id del usuario	Integer	No
Comment	Comentario	Text	No
CreatedAt	Fecha de creación	Timestamp	No
LastUpdatedAt	Fecha de la última actualización	Timestamp	Sí
Claves primarias	Id		
Claves foráneas	SnippetId, UserId		

3.2.5 Integración y entrega continua

Uno de los desafíos en el desarrollo software actual es la llamada automatización de los procesos. Cuando las aplicaciones a desarrollar alcanzan un grado de complejidad importante o el tamaño de los equipos alcanza ciertos valores, resulta crucial garantizar que el software producido en cualquier momento es fiable.

Éste es el objetivo fundamental de la llamada integración y entrega continua¹³. Esta aproximación permite reducir los costes, el tiempo y el riesgo de los proyectos software permitiendo actualizaciones incrementales de las aplicaciones.

Para ello, en el proceso CI/CD se definen diferentes pasos o steps, los cuales representan acciones concretas que permiten la generación de artefactos entregables.

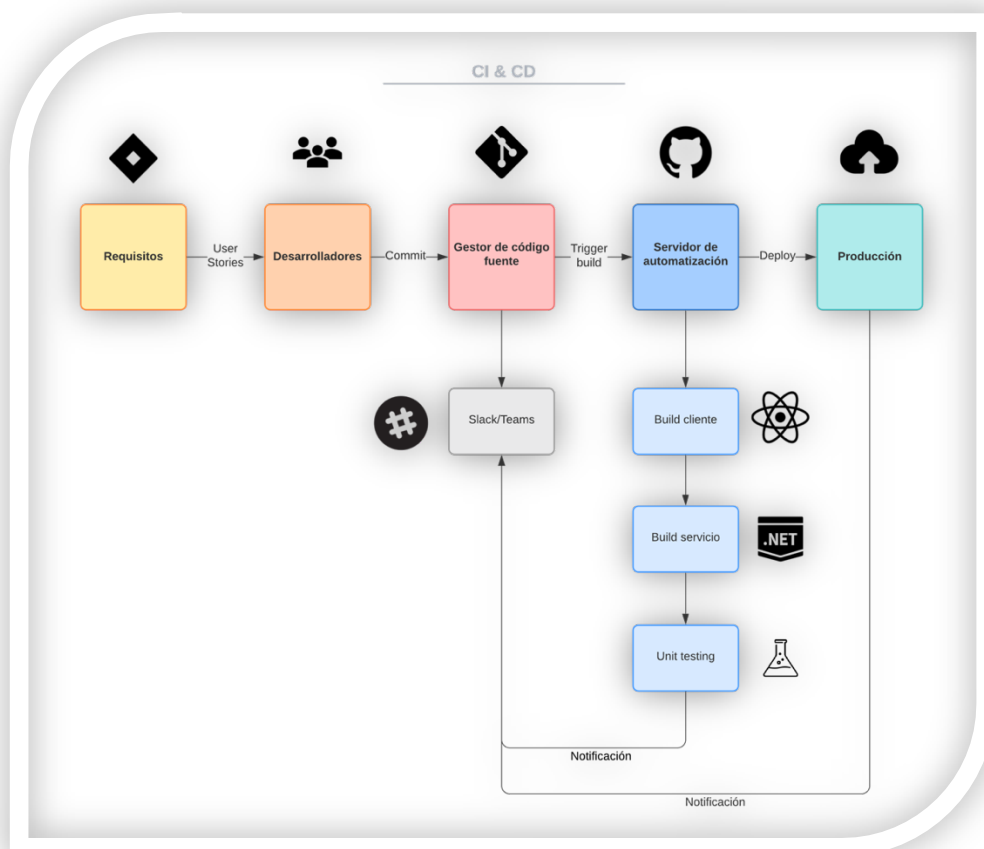


Figura 13: Proceso de integración y entrega continua

¹³ También conocido como "continuous integration and delivery/deployment"

Este proceso es sólo posible mediante la utilización de diferentes herramientas que adicionalmente permiten integraciones entre ellas. Dichas herramientas se describen en el capítulo 1.5.2

Para el proyecto actual y dado que la finalidad es didáctica y no forma parte de sistemas críticos, se ha definido una simplificación de un proceso de desarrollo real¹⁴:

1. Descripción de las tareas en forma de User-Stories en Jira
2. Desarrollo de las diferentes funciones
3. Subidas parciales del código fuente a Git en la rama de desarrollo
4. Fusión del código de la rama de desarrollo con la rama maestra¹⁵ en GitHub. Tras la confirmación se desencadenará en el servidor de automatización un proceso secuencial de tareas o steps que han sido previamente definidos. En nuestro caso:
 - a. Descarga de dependencias y compilación del código cliente (Npm y React)
 - b. Descarga de dependencias y compilación del código servidor (Nuget y .Net Core)
 - c. Ejecución de los diferentes tests unitarios
 - d. Una vez realizados todos los pasos se realizará una notificación a través de un canal de Slack con el estado de las tareas. Cualquier error en cualquier paso abortará el proceso y generará consecuentemente una notificación
5. Despliegue automatizado del cliente y el servidor en aplicaciones preconfiguradas en Azure. Dicho despliegue también generará la respectiva notificación en Slack con el fin de mantener al equipo de desarrollo informado

¹⁴ En procesos reales y más complejos existen más fases y el despliegue se produce mediante triggers manuales por entorno: https://en.wikipedia.org/wiki/Continuous_delivery

¹⁵ Popularmente conocido como Pull-Request: <https://docs.github.com/en/free-pro-team@latest/github/collaborating-with-issues-and-pull-requests/about-pull-requests>

4 Implementación

En el presente capítulo se detallarán las decisiones técnicas tomadas en la fase de implementación del proyecto.

4.1 Visión general

Como ya se ha especificado en anteriores capítulos, el sistema Snippetpedia se ejecutará en una infraestructura cloud creada y configurada específicamente para ello. Bajo este contexto, se diferenciarían tres componentes del sistema:

- **Front-end:** capa de presentación o cliente.
- **Back-end:** capa de servicio, donde reside principalmente la API rest y la lógica de negocio, y que persiste la información en el SGBD.
- **Componentes cloud de infraestructura:** aquellos componentes o herramientas satélite que están presentes en todo proyecto cloud.

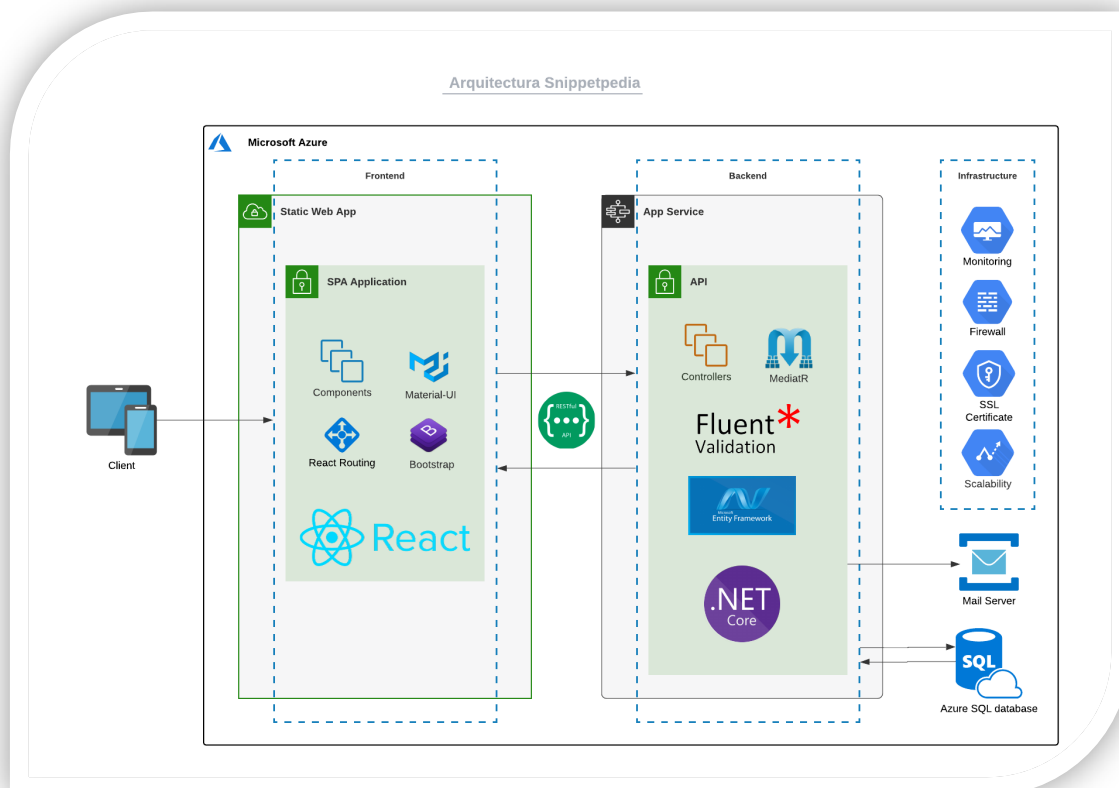


Figura 14: Implementación - Visión general

4.2 Herramientas y librerías

A continuación, se enumeran las diferentes herramientas y librerías que han sido empleadas en el desarrollo del proyecto.

4.2.1 Front-end

La capa de cliente del proyecto ha sido desarrollada con la última versión estable de React (17.0.1) utilizando typescript (4.1.2) y CSS.

React es una biblioteca javascript de código abierto desarrollada y mantenida por Facebook y una extensa comunidad de desarrolladores para construir interfaces de usuario SPA (*single page application*).

El core de react esencialmente permite el desarrollo de componentes, gestionar el estado y ofrece los mecanismos para poder renderizar el estado y modificar el DOM o estructura de la página web que se está visualizando. Pero este *core* no es suficiente para poder desarrollar una nueva aplicación SPA.

Normalmente son necesarios componentes adicionales no incluidos en el *core*, como por ejemplo el enrutamiento.

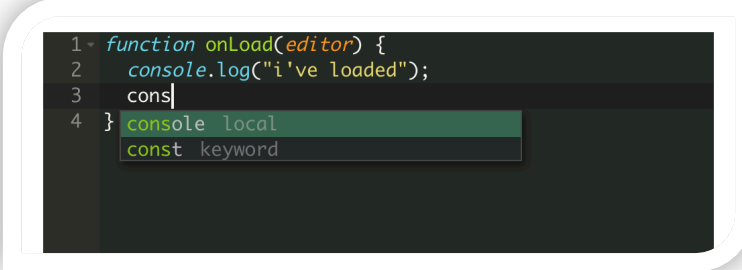
A continuación, pasamos a enumerar aquellas más relevantes, así como la justificación de su elección.

4.2.1.1 Librerías

En el directorio `src/snippetpedia.frontend` se encuentra el fichero `package.json`. Este fichero es utilizado en cualquier proyecto node que mediante el gestor de paquetes NPM, permitirá realizar las descargas de los diferentes paquetes y dependencias.

En el desarrollo del cliente se han empleado las siguientes librerías:

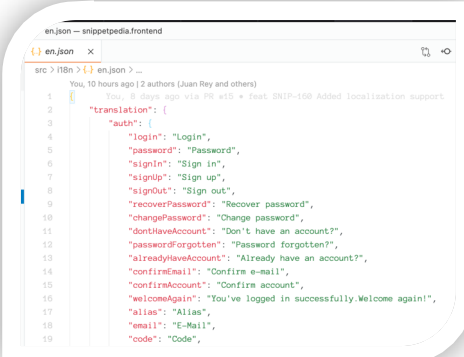
- **Ace (9.2.1):** Editor de código fuente escrito en javascript que puede ser embebido en cualquier aplicación web. Este editor destaca por sus funcionalidades avanzadas y capacidades de personalización. Entre las características a reseñar destaca el soporte de más de 110 lenguajes de programación, resaltado, soporte para temas e *intellisense* (soporte contextual según el lenguaje de programación escogido). Más información: <https://ace.c9.io/>



```
1 function onLoad(editor) {
2   console.log("i've loaded");
3   cons
4 }
  console local
  const keyword
```

Figura 15: Ejemplo de uso del editor Ace

- **Axios (0.21.0)**: cliente HTTP escrito en javascript basado en promesas. Este cliente permite realizar las llamadas a la API Rest e incorpora mecanismos para la gestión de errores. El motivo de la elección ha sido su popularidad y la experiencia previa. Más información: <https://www.npmjs.com/package/axios>
- **Date-fns (2.16.1)**: librería de utilidad empleada para el manejo y formateo de fechas en javascript. Más información: <https://www.npmjs.com/package/date-fns>
- **Dotenv (8.2.0)**: librería de utilidad empleada para la carga dinámica de variables de entorno. En el contexto del proyecto ha sido empleada para la carga de variables como el host, el cual debe ser según entorno configurable. Más información: <https://www.npmjs.com/package/dotenv>
- **I18next (19.8.4)**: soporte para internacionalización de aplicaciones basado en ficheros de recursos como JSON. Más información: <https://www.i18next.com/>



```
en.json - snippetpedia.frontend
src > i18n > en.json > ...
You, 10 hours ago | 2 authors (Juan Rey and others)
You, 8 days ago via PR #15 • feat: SNIP-100 Added localisation support
1 {
2   "translation": {
3     "auth": {
4       "login": "Login",
5       "password": "Password",
6       "signin": "Sign in",
7       "signup": "Sign up",
8       "signout": "Sign out",
9       "recoverPassword": "Recover password",
10      "changePassword": "Change password",
11      "dontHaveAccount": "Don't have an account?",
12      "passwordForgotten": "Password forgotten?",
13      "alreadyHaveAccount": "Already have an account?",
14      "confirmEmail": "Confirm e-mail",
15      "confirmAccount": "Confirm account",
16      "welcomeAgain": "You've logged in successfully.Welcome again!",
17      "alias": "Alias",
18      "email": "E-Mail",
19      "code": "Code",
20    }
21  }
22 }
```

Figura 16: Fichero de internacionalización del proyecto

- **Jest (12.0.0):** framework de testeo unitario en javascript utilizado por otros frameworks como vue o angular. Además de su sencilla configuración, permite generar estadísticas de cobertura de código o simulación de objetos (mocking). Más información: <https://jestjs.io/>
- **Material-UI (4.11.2):** proyecto de código abierto que ofrece componentes React que implementan los patrones y guías de diseño de Google Material Design. Este framework facilita el desarrollo de interfaces mediante el uso de componentes fácilmente configurables que pueden ser incorporados a cualquier aplicación React. Más información: <https://material-ui.com/>

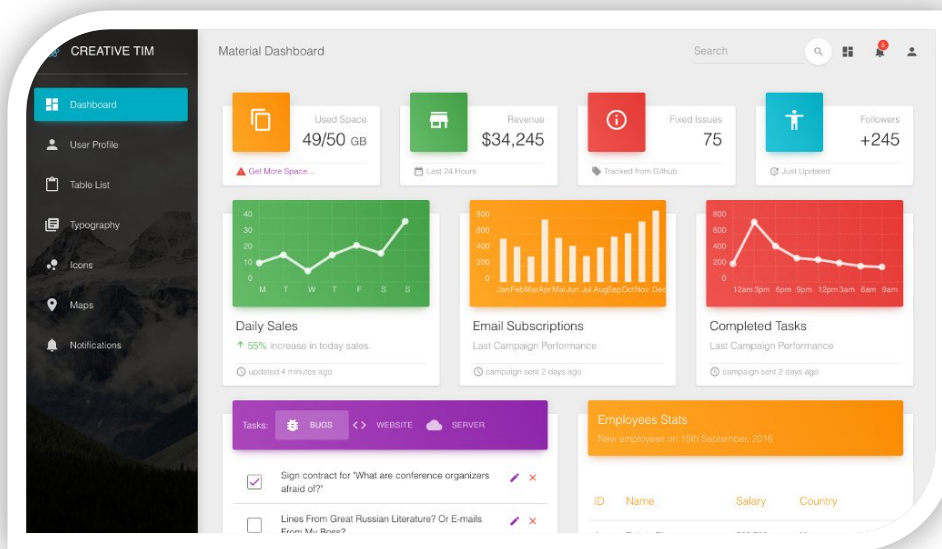


Figura 17: Ejemplo de aplicación basada en Material-UI

- **React-color (2.19.3):** Esta biblioteca permite seleccionar un color de una paleta de colores predefinida. Esta funcionalidad es utilizada en la gestión de los tags, de forma que éstos puedan ser identificados más fácilmente. Más información: <https://casesandberg.github.io/react-color/>

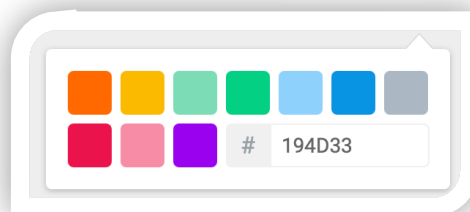


Figura 18: Ejemplo del selector de color react-color

- **React-google-recaptcha (2.1.0):** esta biblioteca permite añadir la funcionalidad *captcha* de Google en el registro de la aplicación, con el fin de evitar la automatización a través de bots. Más información: <https://www.npmjs.com/package/react-google-recaptcha>
- **React-router-dom (5.2.0):** popular y completa biblioteca de enrutamiento para React que mantiene la interfaz de usuario sincronizada con la URL. Más información: <https://reactrouter.com/web/guides/quick-start>
- **React-toast-notifications (2.4.0):** sistema de notificaciones visuales para React. La facilidad de uso así como su interfaz han sido claves en su elección. Más información: <https://jossmac.github.io/react-toast-notifications/>

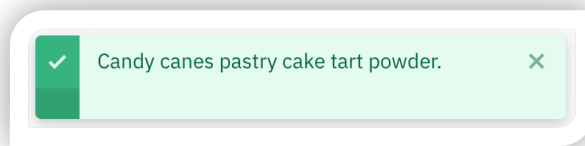


Figura 19: Ejemplo de notificación toast

- **Typescript (4.1.2):** lenguaje de programación desarrollado y mantenido por Microsoft, es un superconjunto de javascript que extiende a éste y es de facto el lenguaje elegido en la inmensa mayoría de nuevos proyectos SPA. Se ha adoptado este lenguaje en lugar javascript “puro” por su popularidad, facilidad y similitud con otros lenguajes de alto nivel. Más información: <https://www.typescriptlang.org/>

4.2.1.2 Herramientas de desarrollo

A continuación, se enumerarán las herramientas de desarrollo empleadas en la implementación del cliente:

- **Nodejs (14.7.0):** empleado para la resolución e integración de dependencias, así como la ejecución de tests unitarios.
- **Visual Studio Code (1.52.1.):** entorno de desarrollo principal con soporte para numerosos lenguajes de programación y extensiones que mejoran la productividad del desarrollador. Algunas de las extensiones más relevantes son:
 - o [AutoCloseTag](#)
 - o [AutoImport](#)

- [ESLint](#)
 - [Prettier](#)
 - [ES7 React Snippets](#)
- **Chrome dev extension:** el propio navegador Chrome incorpora una de las herramientas más completas para desarrolladores para depurar y analizar las aplicaciones web.

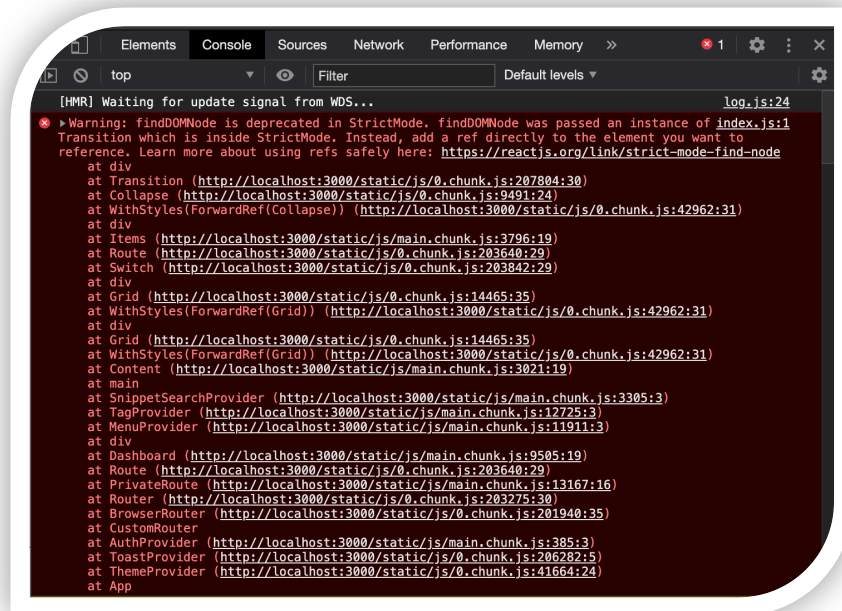


Figura 20: Ejemplo de error con google chrome dev tools

4.2.2 Back-end

La capa de servicio del proyecto ha sido desarrollada con la última versión estable del Microsoft .NET Core Framework (3.1.404).

Este proyecto expone al exterior una serie de funciones REST que posibilitan el intercambio de información con cualquier cliente con dichas capacidades a través de los denominados controladores API.

Estos controladores, basarán su ejecución en el patrón Mediator a través de la librería MediatR. Esta librería permite un bajo acoplamiento a través de su sistema de procesamiento interno basado en mensajes. Estos mensajes serán redireccionados a los respectivos *handlers* o clases encargadas de la lógica de negocio que aplique en cada caso.

Dicha lógica se sustentará por una parte en el framework de validación fluent validator y, en lo referente al acceso a datos, se empleará entity framework, abstrayéndose de la implementación concreta mediante el uso de repositorios de información.

En el siguiente diagrama¹⁶ se resume una implementación clásica basada en este patrón.

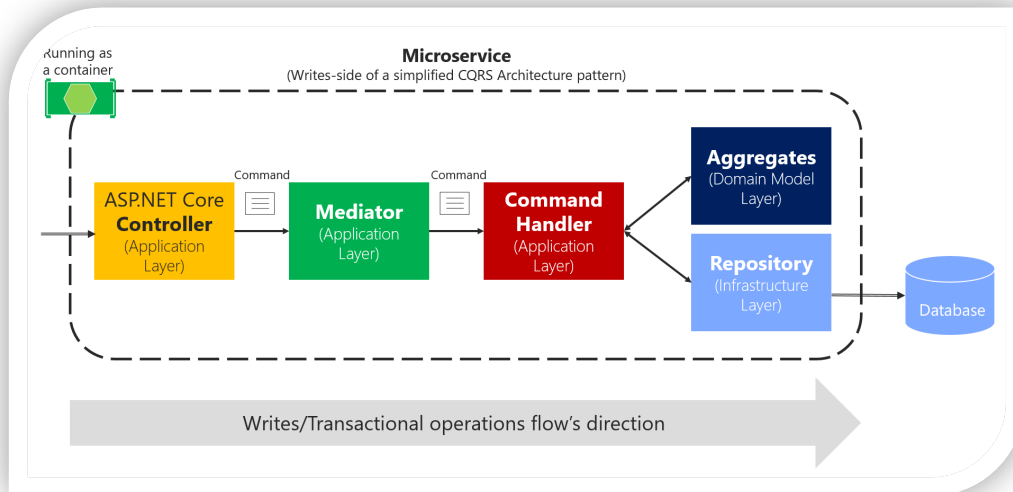


Figura 21: Ejemplo de uso del patrón Mediator

4.2.2.1 Librerías

De forma análoga al cliente, la parte servidora cuenta con un sistema para la descarga y resolución de nuevas dependencias: nuget.

En el proyecto existen actualmente 40 paquetes instalados de repositorios externos, de los cuales la mayor parte son referencias de los paquetes principales. A continuación enumeraremos aquellos más relevantes:

- **FluentMigrator (3.2.9)**: Framework de migración de bases de datos para .NET. Mediante este mecanismo, el sistema en cada inicialización comprobará si hay cambios de base de datos pendientes y las ejecutará. Más información: <https://www.nuget.org/packages/FluentMigrator>.

¹⁶ Ejemplo de utilización del patrón Mediator en un contexto de microservicios: <https://docs.microsoft.com/es-es/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/microservice-application-layer-implementation-web-api>

- **Microsoft.AspNetCore.Authentication.JwtBearer (3.1.10)**: Biblioteca de soporte de tokens JWT, empleados en la fase de autenticación de usuarios. Más información: <https://www.nuget.org/packages/Microsoft.AspNetCore.Authentication.JwtBearer>.
- **Microsoft.EntityFrameworkCore (3.1.10)**: Librería de soporte al ORM Entity Framework de Microsoft. EF soporta múltiples gestores de bases de datos a través de sus diferentes complementos. Más información: <https://www.nuget.org/packages/Microsoft.EntityFrameworkCore>.
- **Moq (4.15.1)**: Popular framework de mocking de para .NET. Mediante este framework es posible realizar tests unitarios sin depender de sistemas externos como una base de datos. Más información: <https://www.nuget.org/packages/Moq>.
- **NLog (4.7.5)**: Librería de soporte avanzado de logging. Más información: <https://www.nuget.org/packages/NLog>.
- **AutoMapper (10.1.1)**: Librería de mapeo de objetos. En el contexto del proyecto es utilizado para mapear entidades de base de datos con los objetos resultantes del servicio. Más información: <https://www.nuget.org/packages/AutoMapper>.
- **FluentValidation (9.3.0)**: Librería de soporte avanzado de validación de objetos con soporte de interfaces fluent¹⁷. Más información: <https://www.nuget.org/packages/FluentValidation>.
- **GlobalExceptionHandler (4.0.2)**: Middleware encargado del mapeo de excepciones a códigos HTTP empleados típicamente en API's Rest. Más información: <https://www.nuget.org/packages/GlobalExceptionHandler>.
- **MailKit (2.10.0)**: Biblioteca con soporte avanzado de envío de correos electrónicos. Más información: <https://www.nuget.org/packages/MailKit>.
- **MediatR (9.0.0)**: Librería con soporte para la implementación de aplicaciones basadas en el patrón Mediator. Más información: <https://www.nuget.org/packages/MediatR>.
- **Swashbuckle (5.6.3)**: Soporte para la documentación de API's basadas en Swagger. Más información: <https://www.nuget.org/packages/Swashbuckle.AspNetCore>.

¹⁷ Fluent interface: https://es.wikipedia.org/wiki/Interfaz_fluida

4.2.2.2 Herramientas de desarrollo

- **Rider (2020.3):** entorno de desarrollo multiplataforma y alternativa a visual studio. Debido a la experiencia previa con este software, se ha utilizado como herramienta principal para la capa de servicio.

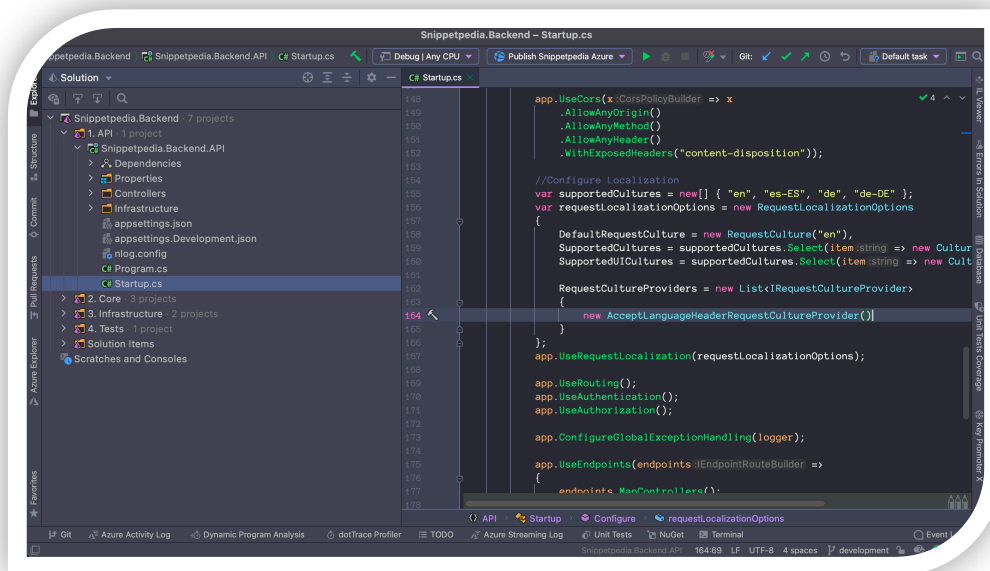


Figura 22: Entorno de desarrollo Rider

- **Datagrip (2020.3):** cliente de gestión y consulta de diferentes servidores de base de datos. La ausencia de un cliente por parte de Microsoft para macOS y las capacidades avanzadas de éste, lo han convertido en la herramienta principal de consulta. Más información:

<https://www.jetbrains.com/datagrip/>

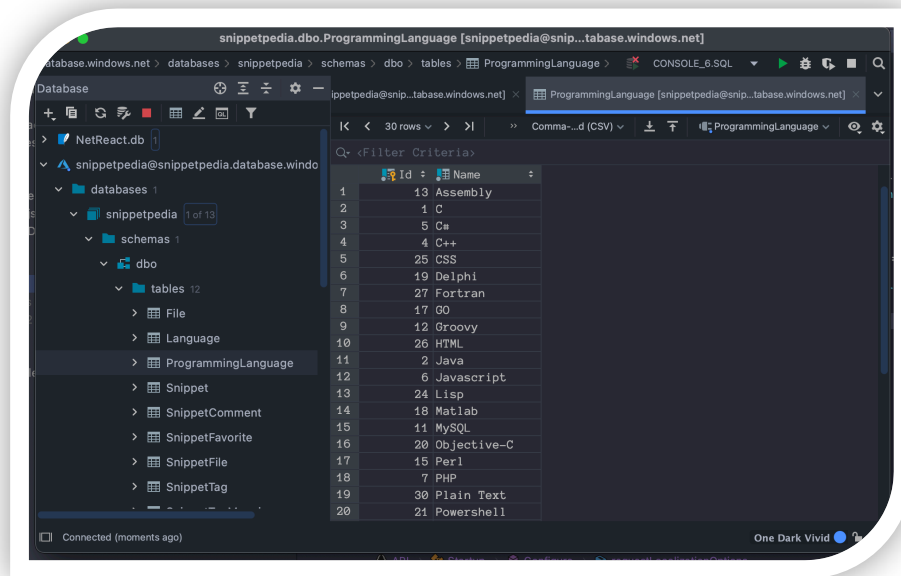


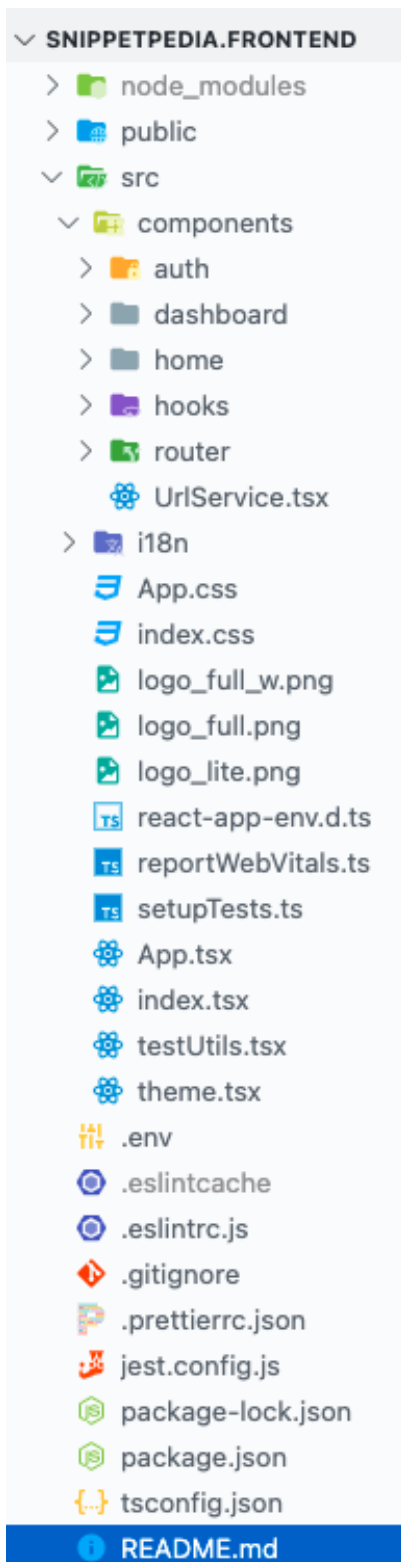
Figura 23: Cliente SGBD Datagrip

- **Postman (7.36.0):** herramienta de consumo de API's rest con funciones de prueba, configuración y scripting. Esta herramienta ha servido en la fase de testeo de las funciones API expuestas por el servicio.

4.3 Estructura del proyecto

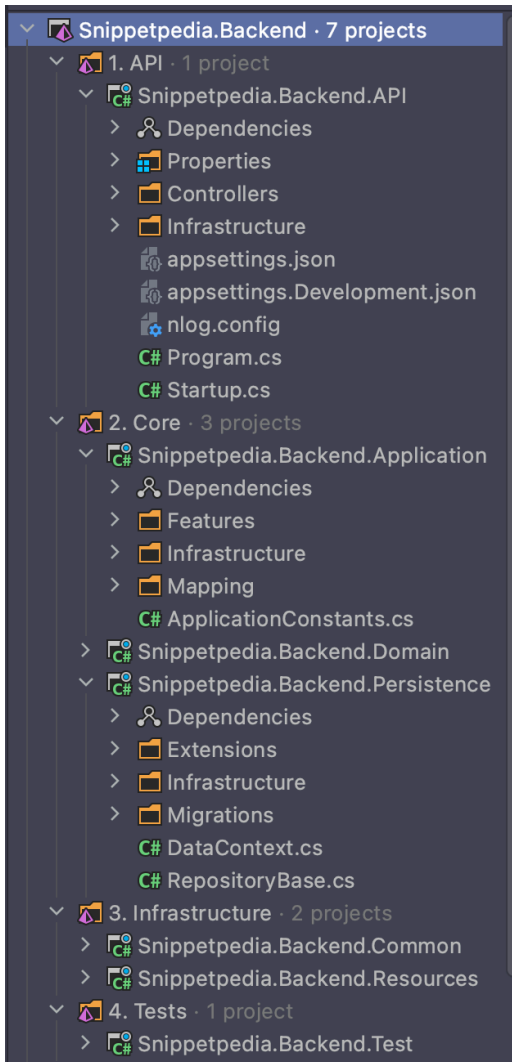
En el presente capítulo se describe la estructura del proyecto de cliente y servidor, así como los principales componentes de los mismos.

4.3.1 Front-end



- **Node_modules:** directorio en el que se almacenan todas las dependencias de terceros gestionadas por npm.
- **Public:** contiene el fichero html básico en el que la aplicación estará embebida
- **Src/components:** React es un sistema fuertemente orientado a componentes, los cuales pueden incluir otros componentes, así como estilos y la lógica de los mismos.
- **Auth:** Contiene todos aquellos componentes relacionados con la autenticación: login, restablecimiento de contraseña, registro y el proveedor de autenticación, utilidad central que proporciona un acceso global de los componentes a características relacionadas con seguridad y usuarios.
- **Dashboard:** Todos aquellos componentes relacionados con los contenidos del dashboard como la barra de búsqueda, el menú de navegación, los snippets resultado de una búsqueda y el propio detalle.
- **Hooks:** Los hooks son un elemento de la nueva API de React que permite que los componentes tengan estado y que éste sea compartido entre todos. En estos directorios se almacenan algunos como el gestor del almacenamiento del navegador, actualización de los diferentes menús o el detalle de un snippet.
- **Router:** Definición de las rutas de la aplicación.
- **I18n:** Ficheros de localización en castellano, inglés y alemán.
- **App.tsx / Index.tsx:** Componentes principales y puntos de entrada de la aplicación.
- **Theme.tsx:** Configuración global del tema del sitio web.
- **.eslintrc:** Fichero de configuración de la guía de estilo de la herramienta ESLint, para mantener la calidad del código.
- **Tsconfig.json:** Configuración de Typescript.
- **Package.json:** Fichero central de dependencias npm.

4.3.2 Back-end



El proyecto backend de Snippetpedia se fundamenta en 4 componentes fundamentales:

- **API:** Componente encargado de la gestión de las peticiones recibidas.
- **Core:** Componentes de la lógica de negocio.
- **Infraestructure:** Componentes transversales de sistema
- **Tests:** Componentes para la ejecución de pruebas unitarias.

A continuación, se describen los sub-proyectos:

- **API:** Proyecto que contiene los diferentes controladores expuestos al cliente final, así como la gestión de la inicialización de la aplicación.
- **Application:** Lógica de negocio basada en consultas y comandos MediatR.
- **Domain:** Objetos de dominio como entidades de base de datos y DTO's (data transfer objects).
- **Persistence:** Estructura de la base de datos fundamentada en migraciones con EF (entity framework).
- **Common:** Librería de utilidades.
- **Resources:** Librería de internacionalización de recursos.
- **Test:** Librería de pruebas unitarias.

4.4 Configuración cloud

Como se ha indicado en capítulos anteriores, el proyecto y sus componentes están alojados en un sistema cloud Azure, el cual, como cualquier sistema de la misma naturaleza, proporciona entre otras, las siguientes características:

- Alta disponibilidad (99,9%¹⁸)
- Seguridad
- Mantenimiento
- Escalabilidad (horizontal y vertical)

Para poder hacer uso de los servicios Azure, en primer lugar, es necesario crear una cuenta en la plataforma. A través de un programa de colaboración entre las universidades y Microsoft, es posible optar a este programa y recibir los créditos correspondientes a 100 dólares y acceso a la mayor parte de los servicios disponibles¹⁹.

La disponibilidad de estos créditos permite tener una suscripción al sistema, en la cual se realizarán los cargos de los diferentes servicios que se estén empleando.

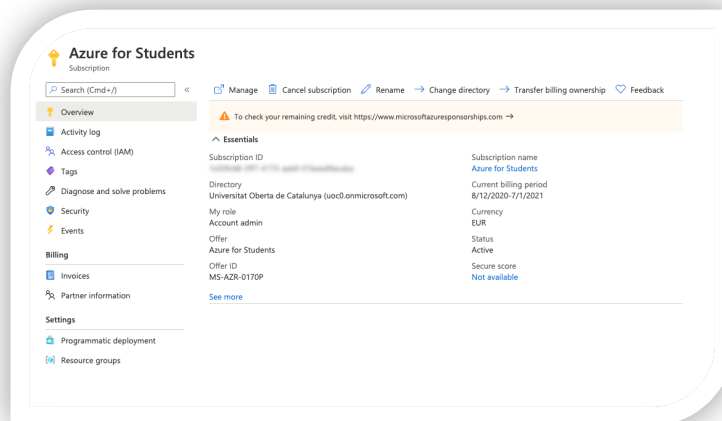


Figura 24: Suscripción a Azure

Por defecto, cualquier recurso en Azure debe pertenecer a los llamados “resource groups” (grupos de recursos). Esto es en esencia una forma de poder agrupar todos los recursos con el fin de poder localizarlos de forma más sencilla y organizada.

¹⁸ Condiciones SLA de Azure: https://azure.microsoft.com/en-us/support/legal/sla/app-service/v1_4

¹⁹ Lista de servicios disponibles con el „free tier“: <https://azure.microsoft.com/en-us/free/free-account-faq>

Para el proyecto Snippetpedia se ha creado a tal fin el grupo de recursos “snippetpedia”, como se puede apreciar en la siguiente imagen:

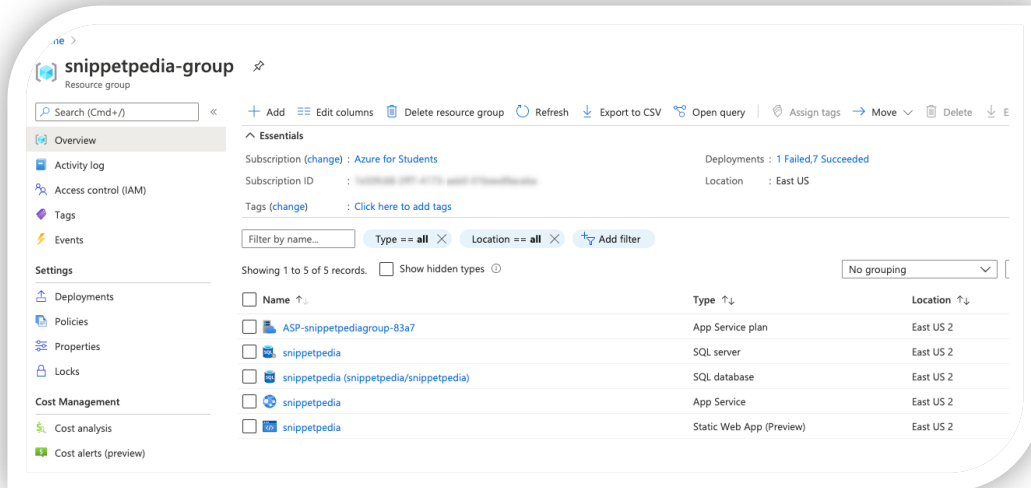


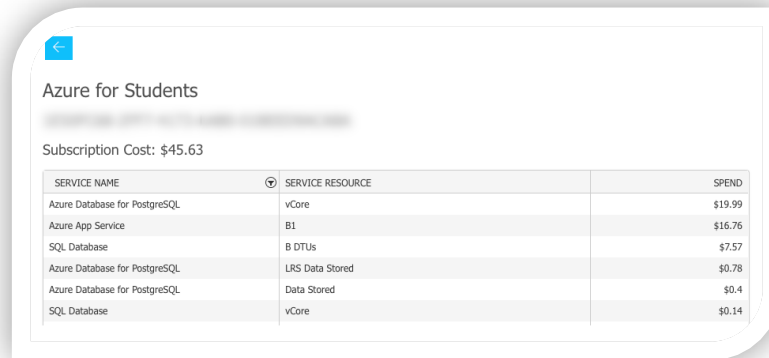
Figura 25: Grupo de recursos en Azure

Como se ha descrito anteriormente (ver Figura 14: Implementación - Visión general), el sistema se compone de cliente, servidor y base de datos. En el contexto cloud en Azure, éstas son las correspondencias con los elementos del grupo de recursos:

- **Cliente:**
 - o Static Web App (Preview): instancia de la aplicación de la parte cliente. Dado que el contenido de la parte cliente es HTML, Javascript y CSS, se considera contenido estático y no es necesario en este caso emplear un App Service.
- **Servidor:**
 - o App Service: instancia de la aplicación de la parte servidora, donde se despliega el código fuente y ensamblados generados, certificados SSL, dominio, etc.
 - o App Service Plan: define el conjunto de recursos de computación de una aplicación web. Este recurso se crea automáticamente con cualquier App Service y gestiona y orquesta los recursos disponibles y los límites de este. Es empleado principalmente para características de escalabilidad.
- **Base de datos:**
 - o SQL Server: servidor de bases de datos SQL Server Azure
 - o SQL Database: base de datos de un servidor SQL Server Azure

Como se puede observar en la imagen anterior, los recursos se pueden albergar en diferentes regiones²⁰. En el proyecto actual se ha elegido la región “East US 2” debido a que los costes de computación son los más económicos (más información: <https://azureprice.net/Region>).

Desde el inicio del proyecto en septiembre de 2020 y con las pruebas realizadas, el coste generado a finales de diciembre asciende a 45\$, de los cuales 21\$ se corresponden con pruebas realizadas anteriormente con otros gestores de base de datos.



SERVICE NAME	SERVICE RESOURCE	SPEND
Azure Database for PostgreSQL	vCore	\$19.99
Azure App Service	B1	\$16.76
SQL Database	B DTUs	\$7.57
Azure Database for PostgreSQL	LRS Data Stored	\$0.78
Azure Database for PostgreSQL	Data Stored	\$0.4
SQL Database	vCore	\$0.14

Figura 26: Costes Azure desde septiembre 2020

4.5 API Rest

La API Rest proporciona una serie de operaciones RESTful para crear, actualizar, eliminar y recuperar entidades u objetos del dominio del sistema. Esta sección proporciona un breve resumen de las características de la API.

Esta API cumple con el nivel 2 de definición del modelo de madurez de Richardson²¹ y se encuentra también disponible en el siguiente enlace: <https://snippetpedia.azurewebsites.net/swagger/index.html>. Asimismo, las respuestas de las funciones retornarán los respectivos códigos HTTP dependiendo del éxito o fracaso de las diferentes operaciones.

/api/Applications/: Gestión de la aplicación		
Ruta	Verbo	Descripción
/myMenu	GET	Devuelve la información necesaria a mostrar en el menú lateral (Contadores, Tags, Lenguajes)

²⁰ Regiones en Azure: <https://azure.microsoft.com/en-us/global-infrastructure/geographies/>

²¹ El modelo de madurez nivel 2 establece el uso de verbos a la hora de realizar las peticiones REST. Más información: <https://restfulapi.net/richardson-maturity-model/>

/api/Files/ : Gestión de ficheros (P.ej: avatares)

Ruta	Verbo	Descripción
/	POST	Crea un nuevo fichero
/{snippetFileId}/	PUT	Actualiza un fichero existente
/{snippetFileId}	DELETE	Elimina un fichero existente
/{snippetFileId}/download	GET	Descarga un fichero existente

/api/ProgrammingLanguages/: Gestión de lenguajes de programación

Ruta	Verbo	Descripción
/	GET	Devuelve todos los lenguajes de programación existentes

/api/Snippets/: Gestión de snippets

Ruta	Verbo	Descripción
/	POST	Crea un nuevo snippet
/{snippetId}	PUT	Actualiza un snippet existente
/{snippetId}	DELETE	Elimina un snippet existente
/{snippetId}	GET	Devuelve un snippet a través de su identificador
/{snippetId}/comments	POST	Crea un comentario en un snippet
/{snippetId}/comments	GET	Devuelve todos los comentarios un snippet existente
/{snippetId}/favorites	POST	Añade un snippet a los favoritos
/{snippetId}/favorites	DELETE	Elimina un snippet de los favoritos
/{snippetId}/files	GET	Devuelve todos los ficheros de un snippet existente
/{snippetId}/fork	POST	Realiza una copia de un snippet público existente como propio
/{snippetId}/votes	POST	Vota un snippet existente
/{snippetId}/votes	DELETE	Elimina el voto de un snippet existente
/{snippetId}/comments/{commentId}	PATCH	Actualiza un comentario existente
/search	POST	Realiza una búsqueda parametrizada sobre los snippets existentes y devuelve los resultados coincidentes

/api/Tags/

Ruta	Verbo	Descripción
/	POST	Crea un nuevo tag
/tagId	PUT	Actualiza un tag existente
/tagId	DELETE	Elimina un tag existente
/tagId/assignment/snippetId	POST	Asigna un tag a un snippet existente
/tagId/assignment/snippetId	DELETE	Elimina la asignación de un tag a un snippet existente
/myTags	GET	Devuelve los tags del usuario actual

/api/Users/

Ruta	Verbo	Descripción
/userId/avatar	GET	Devuelve el avatar de un usuario en base64
/activateUserRegistration	POST	Confirma el registro de un usuario
/authenticate	POST	Autentica a un usuario en el sistema
/avatar	POST	Crea un nuevo avatar para el usuario actual
/avatar	DELETE	Elimina el avatar del usuario actual
/confirmPasswordReset	PATCH	Confirma el restablecimiento de la contraseña
/register	POST	Registra a un nuevo usuario en el sistema
/requestPasswordReset	PATCH	Solicita el restablecimiento de contraseña
/updatePassword	PATCH	Actualiza la contraseña del usuario actual
/updateProfile	PATCH	Actualiza los datos de perfil del usuario actual

5 Demo

El sistema final se encuentra disponible al público en el sitio web www.snippetpedia.com.

5.1 Acceso

Para poder acceder a las funciones de Snippetpedia, es necesario acceder con un usuario registrado. Con el fin de evaluar el sistema existen dos posibilidades:

- Llevar a cabo el registro a través del siguiente enlace: <https://www.snippetpedia.com/register>
- Acceder con un usuario pre-configurado:
 - **Usuario:** demo@snippetpedia.com
 - **Contraseña:** 123.abc!

5.2 Problemas conocidos

En la actualidad no existen problemas conocidos en las funciones existentes.

Por razones de planificación no ha sido posible implementar en el cliente la función que permite editar un comentario de un snippet o votar positiva o negativamente el mismo.

Dichas funciones serán implementadas en un futuro próximo.

6 Pruebas

Durante el desarrollo del proyecto, se han llevado a cabo de forma iterativa y por función las siguientes tareas relacionadas con la fase de pruebas:

- Pruebas de aceptación: por cada función desarrollada, se ha comprobado que la función cumple los requisitos previamente definidos, teniendo en cuenta las siguientes premisas:
 - o El funcionamiento es el esperado
 - o Los contenidos se muestran correctamente en los navegadores más comúnmente usados (Chrome, Firefox, Safari)
 - o No se generan errores inesperados
 - o Los tiempos de respuesta son aceptables
- Pruebas unitarias: las pruebas unitarias permiten con un alto grado de fiabilidad asegurar, nuevos desarrollos o cambios en funciones existentes, no alteran los resultados esperados. Por motivos relacionados con la planificación y, que la mayor parte de la complejidad de la aplicación reside en la capa de servicio, se han desarrollado los correspondientes tests unitarios para cada funcionalidad expuesta. Dicha funcionalidad depende en gran medida de los datos existentes, así que, aprovechando las posibilidades que ofrece Entity Framework a la hora de emplear otros motores de base de datos, se utiliza en tiempo de ejecución de las pruebas una variante de SQLite que permite emplear bases de datos en memoria. En la versión actual se han escrito 48 pruebas unitarias que serán ejecutadas cada vez que se incorpore nuevo código fuente al repositorio.

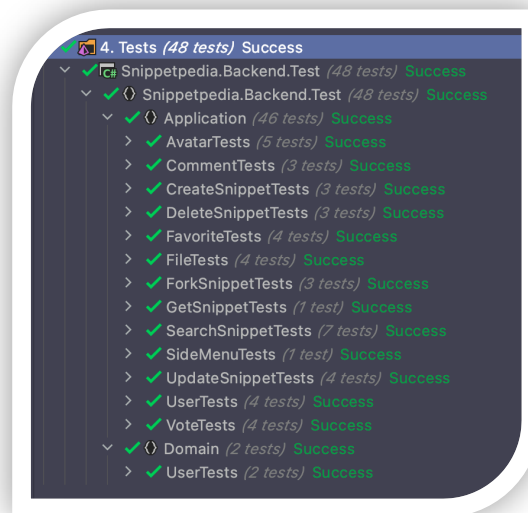


Figura 27: Pruebas unitarias en proyecto back-end

7 Instalación y configuración

Con la entrega de este proyecto y de forma adjunta se incluye el documento **PEC_FINAL_guiaInstalacion_GonzalezRuiz_Carlos.pdf**, que contiene una guía de instalación y configuración de forma local del sistema.

8 Conclusiones

En el presente apartado, y tras la entrega de la primera versión utilizable de Snippetpedia, se detallan brevemente las conclusiones del proyecto.

8.1 Objetivos

El desarrollo de Snippetpedia ha supuesto en algunas fases un desafío importante debido a que se han abordado tareas y actividades con las que no se estaba familiarizado. La elaboración de una memoria y el aprendizaje de nuevas tecnologías han sido un reto considerable, pero en general considero que se han cumplido aquellos objetivos que hace varios meses se habían establecido:

- Se ha creado un producto software que considero de utilidad para cualquier programador.
- Se han respetado los requisitos iniciales y no ha sido necesario recortar funcionalidades.
- Se han adquirido nuevos conocimientos en tecnologías actuales: React y Azure.
- Se ha reforzado la convicción de la importancia de trabajar de una forma estructurada basada en procesos, herramientas y planificación.

8.2 Enseñanzas adquiridas

En el desarrollo de cualquier proyecto o en la propia finalización del mismo, existe una técnica o ejercicio que trata de detectar y analizar aquellos problemas que puedan haber surgido, con el fin de no cometer los mismos errores de nuevo en el futuro. En la metodología SCRUM se conoce como “retrospectiva”, y a nivel general también es conocido como “lessons learned”.

Aunque parece algo obvio y evidente en su formulación, llevarlo a la práctica es muchas veces más complicado de lo que parece, porque implica tener presente todas experiencias pasadas acumuladas en los desarrollos que están por venir, lo cual muchas veces no sucede.

En el proyecto Snippetpedia, se han identificado varios puntos que deberán ser tomados en cuenta en proyectos futuros:

8.2.1 Estimaciones

A pesar de contar con varios años de experiencia en desarrollo software y en distintas tecnologías, las estimaciones en la parte del desarrollo de la parte cliente han sido en algunos casos demasiado optimistas. La curva de aprendizaje en una tecnología que no se domina y que es tan diferente a lo anteriormente conocido ha supuesto un desafío importante.

En este sentido, considero que, los desarrollos futuros con tecnologías nuevas y desconocidas deberán ser más conservadores en lo relativo a las estimaciones y se deberá contar con asesoramiento u opinión experta siempre que sea posible.

8.2.2 Automatización

En varias partes del desarrollo se ha optado por técnicas de automatización y generación de código, con el fin de aprovechar el tiempo de una forma eficiente. En la fase inicial y debido a la necesidad en la fase de implementación de llevar a cabo tareas frecuentes pero muy similares, se llevaron a cabo las siguientes acciones:

- **Desarrollo de cliente**
 - o Generación parcial automática de las estructuras base de los componentes.
- **Desarrollo backend**
 - o Generación parcial automática de las clases de negocio.
 - o Generación parcial de los métodos de controlador API.
- **Despliegues y pruebas**
 - o En cada incorporación a las ramas de desarrollo o principal, se ha configurado Github de forma que cada incorporación de código ejecutará una serie de pruebas en cliente y servidor con el fin de minimizar los errores que se puedan producir²².
 - o La fusión de la rama de desarrollo en la rama principal lanzará los procesos pertinentes que permitirán desplegar el cliente y el servicio automáticamente en Azure.

Estas técnicas han permitido en muchos momentos detectar posibles problemas y han demostrado ser extremadamente útiles.

8.3 Futuro

Snippetpedia no es un producto finalizado cuyo único fin es un proyecto en el ámbito académico.

La idea de Snippetpedia es que pueda ser utilizada en el futuro por otros programadores como herramienta de gestión del conocimiento, teniendo en cuenta la necesidad de incorporar nuevas funcionalidades, que por razones de tiempo no tenían cabida en el proyecto actual. De esta forma se busca que el producto resultante tenga un valor diferenciador con respecto a las herramientas actuales.

Por ello se plantean las siguientes funcionalidades en desarrollos e iteraciones futuras:

²² Estas técnicas ya descrita en anteriores capítulos se conoce también como CI (continuous integration) / CD (continuous delivery)

- **Migración a hosting compartido:** debido a que los costes de servicios cloud son inasumibles en un producto gratuito, se migrará la infraestructura existente a un servicio que sea económicamente sostenible.
- **Búsqueda avanzada de snippets:** con el fin de fomentar la interacción entre usuarios, se plantea mejorar y potenciar la búsqueda de snippets globales añadiendo diversos filtros de búsqueda, que permitan localizar de forma más sencilla y eficiente los resultados coincidentes.
- **Panel de administración:** en la versión actual no existen administradores ni gestión de la configuración. En el futuro, un usuario con los roles adecuados podrá administrar usuarios y configuraciones del sistema.
- **Integración con herramientas de terceros:** la mayor ventaja para un desarrollador que emplee Snippetpedia será la integración con las herramientas que utiliza a diario: el navegador y su entorno de desarrollo favorito (Visual Studio Code, Rider, Eclipse). Debido a que Snippetpedia se basa en tecnologías estándar como REST y JSON, la integración con estas herramientas mediante el desarrollo de los respectivos Plugins, supondrá una ventaja que podrá animar a muchos usuarios potenciales a utilizar Snippetpedia como una herramienta más en su rutina diaria.

9 Bibliografía

- Video-tutoriales de Microsoft sobre Asp.NET Core - <https://dotnet.microsoft.com/learn/aspnet> (septiembre 2020)
- Video-tutoriales de Microsoft sobre Azure - <https://docs.microsoft.com/en-us/learn/azure/> (octubre 2020)
- React – The complete guide (incl. Hooks, React Router, Redux) - <https://www.udemy.com/course/react-the-complete-guide-incl-redux/> - Maximilian Schwarzmüller (Octubre 2020)
- Material-UI – <https://material.ui.com> (septiembre 2020 – enero 2020)
- Reactjs – <https://reactjs.org> (septiembre 2020 – enero 2020)
- React router – <https://reactrouter.com> (septiembre 2020 – enero 2020)
- Typescript reference - <https://www.typescriptlang.org/> (septiembre 2020 – enero 2020)
- Eslint reference - <https://eslint.org/> (octubre 2020)
- Axios reference - <https://github.com/axios/axios> (noviembre 2020)
- MediatR - <https://github.com/jbogard/MediatR> (octubre 2020)
- Github actions - <https://github.com/features/actions> (septiembre 2020)
- Stackoverflow developer survey - <https://insights.stackoverflow.com/survey/2020> (noviembre 2020)