

# Memoria del proyecto

*Diseño de página Web para aprendizaje de idiomas.*

Trabajo de Fin de Grado

INGENIERÍA INFORMÁTICA



**VNiVERSiDAD  
D SALAMANCA**

Diciembre de 2022

**Autor:**

*Angel Cantuche Martín*

**Tutores:**

*Gabriel Villarrubia González*

*André Filipe Sales Mendes*

## Certificado del/los tutor/es TFG

D./Dña. André Filipe Sales Mendes y Gabriel Villarrubia González,  
profesores del Departamento de informática de la Universidad de Salamanca,

HACE/N CONSTAR:

Que el trabajo titulado  
“MYLEARNING: aplicación para autoaprendizaje de idiomas”, que se presenta, ha  
sido realizado por Angel Cantuche Martin con DNI 71038039D y constituye la  
memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de  
Grado en ingeniería informática en esta Universidad.

Salamanca, 12 de enero de 2022

Fdo.: \_\_\_\_\_

Fdo.: \_\_\_\_\_



## Resumen

Este proyecto de diseñar una página web para aprender idiomas, en este caso inglés, va de la mano del auge que tiene el conocimiento de idiomas actualmente, junto con oportunidades que emergen de países extranjeros, ya que actualmente se da una alta relevancia o inclinación por aquellas personas que tienen un alto nivel de manejo del idioma, haciendo decantar incluso la balanza en una situación igualitaria por estas personas.

La página web puede ser usada por personas de todas las edades con cualquier nivel. Puede haber opciones donde una propia persona sea autodidacta y en aquellas situaciones donde alguien quiera usarlo de recurso para apoyarse o facilitar al “alumnado” una extra de dinámicas.

Por parte del desarrollo ha querido montarse de una manera sencilla y fácil de navegar, puesto que se enfoca desde un rango de infancia hasta uno de edad avanzada.

**Palabras clave:** idiomas, inglés, dinámicas.



## Summary

This project of designing a website to learn languages, in this case English, goes hand in hand with the boom that has the knowledge of languages currently, along with opportunities that emerge from foreign countries, as currently there is a high relevance or inclination for those people who have a high level of language management, making even tip the balance in an equal situation for these people.

The website can be used by people of all ages with any level. There may be options where a person is self-taught and, in those situations, where someone wants to use it as a resource to support or provide the "students" with extra dynamics.

On the development side, it has been designed in a simple and easy to navigate way, since it is focused from a range of childhood to an advanced age.

Keywords: languages, English, dynamics.



## Tabla de contenido

|                                         |                                             |
|-----------------------------------------|---------------------------------------------|
| <b>CERTIFICADO DE LOS TUTORES .....</b> | <b><i>¡Error! Marcador no definido.</i></b> |
| <b>Resumen .....</b>                    | <b>4</b>                                    |
| <b>Summary.....</b>                     | <b>6</b>                                    |
| <b>1. Introducción.....</b>             | <b>14</b>                                   |
| <b>2. Objetivos .....</b>               | <b>15</b>                                   |
| 2.1 Objetivos primarios .....           | 15                                          |
| 2.2. Objetivos personales .....         | 16                                          |
| <b>3. Conceptos teóricos.....</b>       | <b>17</b>                                   |
| 3.1 Página Web .....                    | 17                                          |
| 3.2. ¿Qué es WWW? .....                 | 18                                          |
| 3.3. ¿Qué es HTTP? .....                | 19                                          |
| 3.4 Modelo Cliente – Servidor .....     | 19                                          |
| 3.5 Hosting .....                       | 20                                          |
| 3.6 Framework.....                      | 21                                          |
| 3.7 Frontend y Backend.....             | 21                                          |
| 3.6 Librerías o bibliotecas .....       | 21                                          |
| <b>4. Técnicas y herramientas .....</b> | <b>22</b>                                   |
| 4.1 Entornos de desarrollo.....         | 22                                          |
| 4.1.1 Visual studio Code .....          | 22                                          |
| 4.2 Framework .....                     | 22                                          |
| 4.2.1 Vue.js .....                      | 22                                          |
| 4.2.2 Vuetify.....                      | 23                                          |
| 4.3 Librerías.....                      | 23                                          |
| 4.3.1 Firebase.....                     | 23                                          |
| 4.4 Lenguajes .....                     | 23                                          |
| 4.4.1 UML.....                          | 23                                          |
| 4.4.2 HTML.....                         | 24                                          |
| 4.4.3 CSS .....                         | 24                                          |
| 4.4.5 JavaScript .....                  | 25                                          |

|                                                       |           |
|-------------------------------------------------------|-----------|
| 4.4.6 JSON.....                                       | 26        |
| 4.5     Hosting y herramientas de compilación .....   | 26        |
| 4.5.1   Node.js .....                                 | 26        |
| 4.5.2   Npm .....                                     | 27        |
| 4.6     Herramientas de control de versiones.....     | 27        |
| 4.6.1   GitHub .....                                  | 27        |
| 4.7     Herramientas CASE .....                       | 27        |
| 4.7.1   Visual Paradigm .....                         | 27        |
| 4.7.2   Microsoft Project .....                       | 27        |
| 4.7.3   EZEstimate.....                               | 27        |
| 4.8     Herramientas de documentación .....           | 28        |
| 4.8.1   TypeDoc .....                                 | 28        |
| Figura 14. Ejemplo 9 TypeDoc .....                    | 28        |
| <b>5     Aspectos relevantes del desarrollo .....</b> | <b>29</b> |
| 5.1     Marco de trabajo .....                        | 29        |
| 5.2     Estimación de coste y esfuerzo .....          | 31        |
| 5.3     Planificación temporal .....                  | 32        |
| 5.4     Especificación de requisitos .....            | 36        |
| 5.4.1   Participantes.....                            | 36        |
| 5.4.2   Objetivos .....                               | 37        |
| 5.4.3   Actores .....                                 | 38        |
| 5.4.4   Diagrama de paquetes .....                    | 39        |
| 5.5.6   Requisitos de información .....               | 41        |
| 5.5.7   Requisitos no funcionales .....               | 42        |
| 5.5     Análisis de requisitos .....                  | 42        |
| 5.5.1   Modelo de dominio .....                       | 42        |
| 5.5.2   Análisis de requisitos .....                  | 44        |
| 5.6.3   Clases de análisis .....                      | 45        |
| 5.6     Diseño del sistema .....                      | 45        |
| 5.6.1   Patrones arquitectónicos .....                | 45        |
| 5.6.2   Subsistema del diseño .....                   | 46        |

|                                                               |           |
|---------------------------------------------------------------|-----------|
| 5.6.2.1 Subsistema de componentes .....                       | 46        |
| 5.6.2.2 Realización de casos de uso .....                     | 47        |
| 5.6.2.3 Descripción de viista arquitectónica .....            | 47        |
| 5.6.2.4 Modelo de despliegue .....                            | 48        |
| <b>6. Implementación .....</b>                                | <b>48</b> |
| <b>7. Pruebas .....</b>                                       | <b>49</b> |
| <b>8. Aplicación.....</b>                                     | <b>50</b> |
| 8.1 Vista principal .....                                     | 50        |
| 8.2 Acceso a la aplicación .....                              | 51        |
| 8.3 Perfil del usuario .....                                  | 52        |
| 8.6 Recursos .....                                            | 53        |
| 8.5 Listado de actividades / realización de actividades ..... | 55        |
| 8.6 Contacto.....                                             | 57        |
| 8.6 Cerrar sesión .....                                       | 58        |
| <b>9 Conclusiones y líneas futuras .....</b>                  | <b>59</b> |
| 9.1 Conclusiones.....                                         | 59        |
| 9.2 Líneas de trabajo futuras .....                           | 60        |
| <b>10 Bibliografía .....</b>                                  | <b>61</b> |

## Tabla de figuras

|                                                               |    |
|---------------------------------------------------------------|----|
| Figura 1. Página web .....                                    | 17 |
| Figura 2. Página Web sin estilos .....                        | 17 |
| Figura 3. DNS .....                                           | 18 |
| Figura 4. Modelo Cliente-Servidor.....                        | 19 |
| Figura 5. Arquitectura Cliente – Servidor.....                | 20 |
| Figura 6. Hosting – Servidor .....                            | 21 |
| Figura 7. Visual Studio Code.....                             | 22 |
| Figura 8. Código HTML .....                                   | 24 |
| Figura 9. Código CSS.....                                     | 24 |
| Figura 10. Página Web sin CSS.....                            | 25 |
| Figura 11. Página web con CSS .....                           | 25 |
| Figura 12. Código JavaScript.....                             | 26 |
| Figura 13. Ejecución código JavaScript.....                   | 26 |
| Figura 15. Proceso Unificado .....                            | 29 |
| Figura 16. Iteraciones Proceso Unificado.....                 | 30 |
| Figura 16. Resultado de estimación de coste y esfuerzo .....  | 31 |
| Figura 17. Proceso unificado mpp .....                        | 32 |
| Figura 20. Actores .....                                      | 38 |
| Figura 22. Diagrama de paquetes.....                          | 39 |
| Figura 23. Diagrama paquete gestión de usuarios. ....         | 39 |
| Figura 27. Modelo de dominio .....                            | 43 |
| Figura 28. Análisis de requisito – diagrama de secuencia..... | 44 |
| Figura 29. Clases de análisis - diagrama de comunicación..... | 45 |
| Figura 30. MVC .....                                          | 46 |
| Figura 31. Diagrama Components .....                          | 46 |
| Figura 32. Diagrama de secuencia teoría .....                 | 47 |
| Figura 28. Descripción de vista arquitectónica.....           | 47 |
| Figura 33. Descripción de vista arquitectónica.....           | 48 |
| Figura 34. HomePage .....                                     | 50 |
| Figura 35. Login.....                                         | 51 |
| Figura 36. Registro .....                                     | 52 |
| Figura 37. Ver perfil .....                                   | 52 |
| Figura 38. Modificar perfil.....                              | 53 |
| Figura 39. Menú desplegable .....                             | 53 |
| Figura 40. Verbos irregulares .....                           | 54 |
| Figura 41. Recetas inglesas.....                              | 54 |
| Figura 42. Recurso festividades inglesas.....                 | 55 |

|                                               |           |
|-----------------------------------------------|-----------|
| <b>Figura 43. Niveles de inglés .....</b>     | <b>55</b> |
| <b>Figura 44. Listado de actividades.....</b> | <b>56</b> |
| <b>Figura 45. Corregir actividades .....</b>  | <b>56</b> |
| <b>Figura 46. Mostrar soluciones .....</b>    | <b>57</b> |
| <b>Figura 47. Icono contactar.....</b>        | <b>57</b> |
| <b>Figura 48. Formulario contactar .....</b>  | <b>58</b> |
| <b>Figura 49. Logout .....</b>                | <b>58</b> |

**Indice de tablas**

**Tabla 1. Autor 1 ..... 37**  
**Tabla 2. Objetivo 001 ..... 37**  
**Tabla 3. Actor 0001 ..... 38**  
**Tabla 4. UC-0001..... 40**  
**Tabla 5. IRQ 0001..... 41**  
**Tabla 6. NFR-0001 ..... 42**

## 1. Introducción

Como se ha hablado antes, la posibilidad de mejorar en el manejo de un idioma como es el inglés actualmente tiene un gran peso. Tanto profesional como personalmente.

Alguna de las ventajas que tiene el saber el idioma pasan por aumentar las oportunidades de conseguir trabajo como por tener mayor seguridad de cara a mantener un trato con personas extranjeras o viajar a dichos lugares.

Aprender inglés no solo consiste en aprender estructuras gramaticales y vocabulario, sino entonación, lenguaje, léxico y jerga...

El inglés es el idioma de internet y la tecnología. Para entender y comunicarte con la mayor parte del mundo, el inglés es posiblemente la herramienta más importante.

Hoy en día la tecnología avanza más rápido que el tiempo y muchas veces saltar esta brecha del idioma es muy importante porque más que nunca en las empresas se trabaja de forma multicultural, y poder comunicarte con compañeros es algo esencial.

En este documento se explicarán los aspectos más importantes en el desarrollo del proyecto. La estructura será la siguiente:

1. **Objetivos:** se especifican los objetivos que tiene este proyecto.
2. **Conceptos teóricos:** se detallarán los conceptos teóricos para una comprensión más clara del proyecto.
3. **Técnicas y herramientas:** documentación de las técnicas para diluir posibles dudas y herramientas usadas que se han llevado a cabo en este proyecto.
4. **Aspectos relevantes del desarrollo:** aspectos más relevantes en el desarrollo.
5. **Conclusiones:** se especificarán las conclusiones obtenidas como consecuencia de la realización del proyecto.
6. **Bibliografía**

Este documento está completado con los siguientes anexos:

- **Anexo I – Planificación temporal.**
- **Anexo II – Especificación de requisitos.**
- **Anexo III – Análisis de requisitos.**
- **Anexo IV – Documentación técnica**
- **Anexo V – Manual de usuario**

## 2. Objetivos

Aquí reflejaremos los objetivos que cumplirá el sistema propuesto en este Trabajo Fin de Grado, desarrollaremos dichos objetivos y añadiremos la motivación personal a la hora de enfocar este tema.

### 2.1 Objetivos primarios

El objetivo base de este Sistema es el desarrollo de una página web para la mejora del idioma extranjero, en el caso, el inglés. Este sistema permitirá la realización de ejercicios a distintos niveles tratando de un nivel gramatical o de vocabulario, a comprensión escrita.

Como se han descrito en el *Anexo II – Especificación de requisitos*, los objetivos que el sistema deberá cumplir son los siguientes:

- **Interacción del usuario con el sistema:** El sistema deberá ser fácil e intuitivo para que cualquier persona con el rango de edad que tenga pueda navegar de una manera sencilla por la página, sabiendo en todo momento a que tipo de ejercicio accede y los pasos que debe seguir para que éste sea resuelto.
- **Presentación y visualización de los recursos del sistema:** Para que el objetivo anterior sea resuelto, el sistema deberá mostrar la información de la página con los distintos recursos, teniendo una disponibilidad inmediata de los mismos.
- **Corrección de respuestas:** el sistema será capaz de corregir las respuestas informadas por el usuario, contabilizando los fallos y aciertos que haya

tenido en una escala hasta  $n$ , siendo  $n$  el número de preguntas que contenga dicho ejercicio.

- **Muestra de las soluciones a los ejercicios:** el sistema debe ser capaz de mostrar las soluciones de los ejercicios, al margen de la puntuación, tendrá una opción dónde el usuario podrá ver las soluciones del ejercicio sin necesidad de repetirlo infinitas veces para tener las respuestas bien.
- **Modificación de actividades:** El sistema mostrará las modificaciones de los ejercicios, así como la creación, borrado o alteración por errores de ellos.

## 2.2. Objetivos personales

Lo siguiente que veremos son los objetivos personales o motivación propia en la realización de este Trabajo Fin de Grado.

En este proyecto he podido desarrollar la parte Front-End que hay en las aplicaciones, es algo que a mí me llama la atención, el diseño y la estructuración de los componentes es algo tan relevante como llamativo al uso.

El atractivo y simplicidad de una página hace que la experiencia del usuario sea mucho más favorable. Cuando haces que el usuario se fije en los que de verdad importa y quitas todo tipo de despistes, es cuando la experiencia de este es más positiva, mejorando la usabilidad de la aplicación

Claro que esta, que la parte anterior está respaldada del Back-End, utilizando bases de datos para el almacenaje de la gestión de usuarios, de las actividades, y de tareas propias de la página, como, por ejemplo, un recuento de las actividades que ha realizado cada usuario y el progreso de este.

### 3. Conceptos teóricos

En este apartado hablamos de los conceptos teóricos que una persona ajena a la tecnología por completo le pueda causar dudas.

#### 3.1 Página Web

Una Página Web, o página de Internet coloquialmente, es un documento electrónico visualizado en la Web.



Figura 1. Página web

Compuesta primordialmente por información, agrupando dentro de ello textos, imágenes, vídeos o cualquier otro recurso adaptado al sitio Web.

Todo ello mostrado con cierto estilo para el ojo de la persona y su propia facilidad de entendimiento. Ese documento puede tener enlaces *hipervínculos* que dan la capacidad de usar de manera interactiva la página Web. Un ejemplo sin estilo o sin *css*, que es el lenguaje comúnmente usado para ello es el de la figura 2.



Figura 2. Página Web sin estilos

Las páginas Web son implementadas o escritas en lenguajes apropiados para ello, el que ahora brinda mayores ventajas y más comúnmente conocido es el lenguaje HTML, el cuál hablaremos posteriormente.

Como dato extra las páginas Web se dividen en páginas Web estáticas o generado al momento de visualizarlas y páginas Web dinámicas dónde dicha página solicita su contenido a un servidor Web.

### 3.2. ¿Qué es WWW?

Las iniciales **World Wide Web**, son los documentos de hipertexto a los que cada persona puede acceder mediante un navegador a través de Internet.

Su funcionamiento es sencillo, en el momento que un usuario escribe una dirección (URL) en su navegador, este inicia una búsqueda de la misma, mostrando por pantalla los datos requeridos o en su defecto los que encontró relacionado con ello.

Internamente y como hemos visto en asignaturas del Grado de Ingeniería Informática, lo primero que hace es transformar ese nombre literal que a ingresado el usuario en una dirección IP, o bien la tenía ya registrada por haber accedido anteriormente o si no irá a consultar a un/os servidor/es de nombres (DNS). En Figura 3 podemos ver el diagrama básico de comunicación con el dispositivo del usuario.

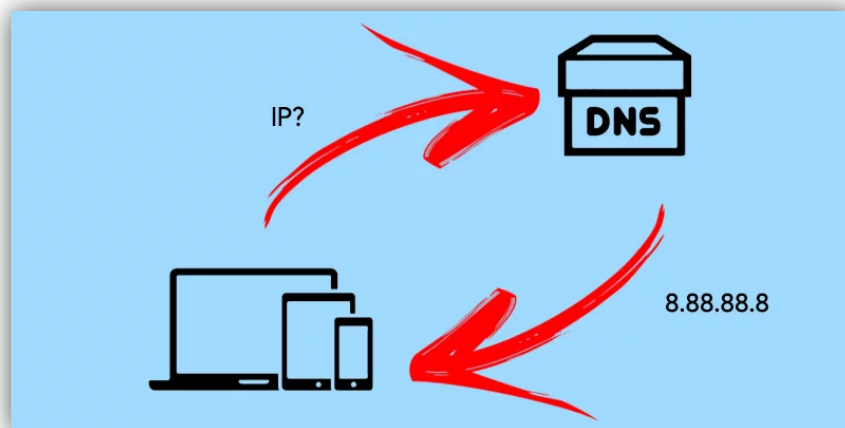


Figura 3. DNS

Con la dirección IP el navegador comenzará la transferencia de paquetes, las solicitudes de HTTP al servidor logran finalmente la visualización del HTML.

Vale la pena destacar que por todo ello el mundo se ha transformado en la auténtica revolución no solo tecnológica, sino también social.

### 3.3. ¿Qué es HTTP?

**Hypertext Transfer Protocol** es el protocolo de comunicación que brinda la posibilidad de intercambio de información a través de WWW.

A vista de pájaro el HTTP define la sintaxis y semántica que emplea la arquitectura WWW, a través de peticiones de un cliente contra un servidor, aunque este esquema puede cambiar de rol, ya que hay peticiones clientes a clientes o servidores a servidor y en un momento dado puede intercambiar su rol. En la figura 4 describimos el esquema básico de conexión con los terminales.

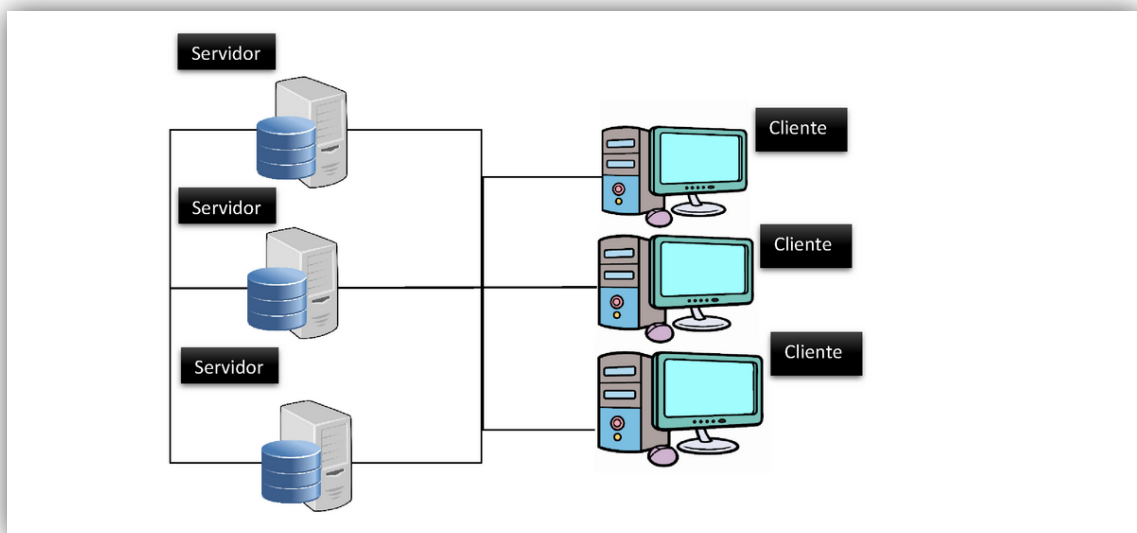


Figura 4. Modelo Cliente-Servidor

Estas peticiones no son más que mensajes con un determinado formato, HTTP es el encargado de determinar los métodos de las peticiones. El servidor retorna un código como respuesta a estos mensajes.

### 3.4 Modelo Cliente – Servidor

Ha salido más de una vez la expresión Cliente – Servidor, vamos a desglosar de qué se trata.

Por un lado, se dice que un dispositivo es un cliente cuando requiere unos recursos o servicios por parte de un servidor.

Por otro lado, se llama servidor al dispositivo que brinda los recursos o servicios al cliente mediante una red. En figura 5 podemos ver globalmente el funcionamiento básico de cliente-servidor.

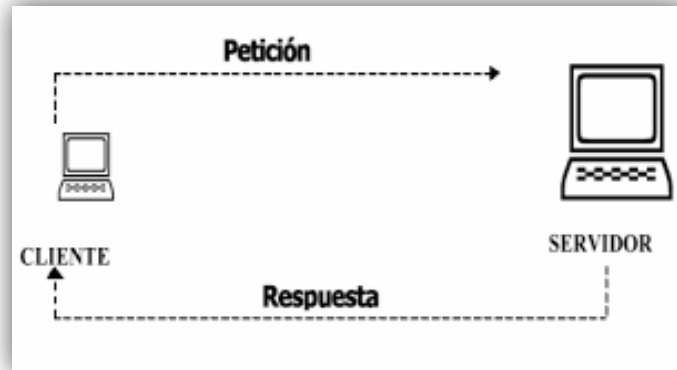


Figura 5. Arquitectura Cliente – Servidor

Esta arquitectura posibilita a distribuir la carga entre distintos servidores para las demandas de múltiples clientes.

### 3.5 Hosting

Hosting o también conocido como alojamiento Web es el servicio o la “caja” donde se almacena una página Web, en su defecto, los datos de esta.

De la mano de esto y que se ha hablado antes es el denominado servidor, un hosting es un servidor el cual genera la respuesta de las peticiones de los clientes.

Cabe destacar que hoy en día se pueden encontrar números alojamientos Web, cada uno con distinta peculiaridad. Desde hosting gratuitos que ofrecen características limitadas como bajo número de peticiones de usuarios, bajo almacenamiento y poca eficiencia, hasta hosting con gran capacidad tanto de almacenamiento como de comportamiento ante peticiones de usuarios, y posibilidad de mantener más de una base de datos con alta seguridad.

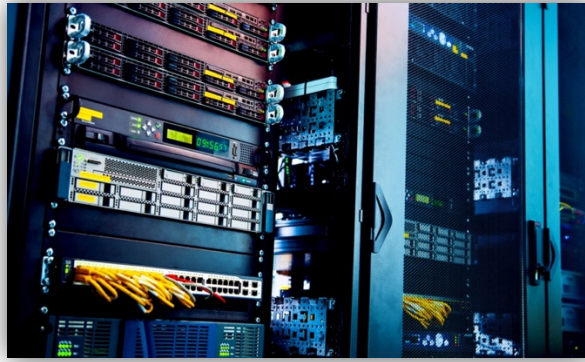


Figura 6. Hosting – Servidor

### 3.6 Framework

Es un marco de trabajo que se utiliza de base para la implementación y desarrollo de un proyecto de software. Posibilita una más rápida programación, ya que implementa internamente códigos nativos para que resulte más sencilla esta tarea, además, de dotar de una mayor organización y estructura al código.

### 3.7 Frontend y Backend

A la hora de desarrollar una aplicación podemos ver dos lados diferenciados. El frontend y el backend. El primero representa la parte visual dónde navega el usuario, y la segunda representa la base de datos donde se almacenan los datos, es la parte transparente al usuario.

### 3.6 Librerías o bibliotecas

Una librería o biblioteca es aquel soporte que dota de una funcionalidad extra ya desarrollada a una aplicación, esta librería se incluye a nivel de código en la implementación del sistema, listas para usar, probadas con anterioridad y fáciles de acoplar.

## 4. Técnicas y herramientas

### 4.1 Entornos de desarrollo

#### 4.1.1 Visual studio Code

Visual Studio Code es un entorno de programación que está dotado con una interfaz más amigable para la persona que desarrolla el software, estructura y organiza el proyecto con un árbol adecuado para su correcta modularidad, fácil entendimiento y mantenimiento en el futuro. Además de todo esto, este entorno de desarrollo nos da la oportunidad de desarrollar con una mayor claridad de código, de una manera más rápida y con el añadido de poder obtener una terminal dentro del propio IDE(entorno de desarrollo integrado).

A su vez, Visual Studio Code nos permite clonar un repositorio Git de control de versiones para poder ir subiendo a él cada actualización del desarrollo.

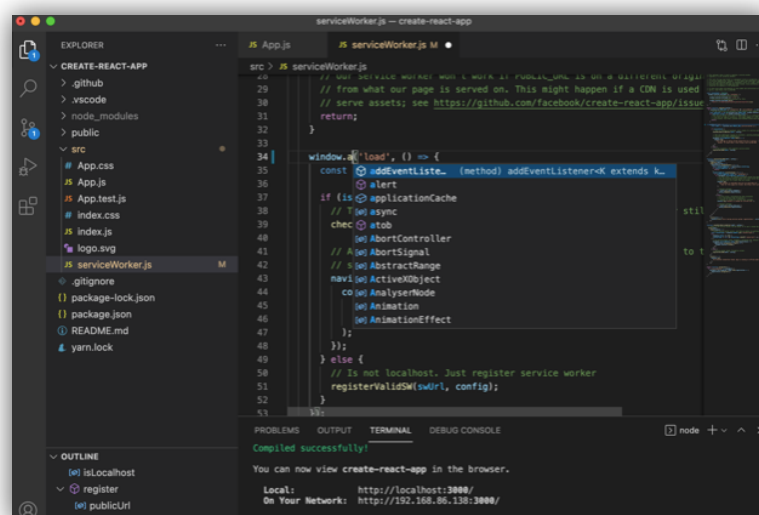


Figura 7. Visual Studio Code

### 4.2 Framework

#### 4.2.1 Vue.js

Vue.js es un framework de JavaScript, en el cual podemos implementar diversos desarrollos de una manera fácil, implementar interfaces de usuarios de forma sencilla y generar aplicaciones totales y funcionales.

Con este framework se pueden implementar todas las vistas de la aplicación, acoplarlo a un backend para que utilice una base de datos, y desde lo más básico a lo más complejo.

En su valor añadido esta la rápida curva de aprendizaje que tiene, ya que no conlleva mucho esfuerzo para entender y aprender esta tecnología, esto conlleva a una positiva experiencia del desarrollador.

#### 4.2.2 Vuetify

Vuetify es un framework que nos permite la creación de vistas de usuarios, combina Vue.js + Material Design. Incluye diversos elementos como botones, sliders, snackbars, cards... debido a esto tiene un rápido desarrollo de software, todos estos componentes se pueden personalizar con distintas opciones para cambiar en aspecto y estética, además de poder configurar también el comportamiento de este.

Vuetify.js es soportado por múltiples navegadores, Safari, Chrome, Edge, Firefox...

### 4.3 Librerías

#### 4.3.1 Firebase

Es una plataforma creada por Google para desarrollar y facilitar la creación de aplicaciones. Contiene funcionalidades típicas que un desarrollador suele implementar, además de poder acoplar en nuestro sistema su base de datos, en mi caso para la gestión de usuarios.

Se encuentra alojada en la nube, y por tanto, es multiplataforma. A su vez, cuenta con distintas funciones para que cualquiera pueda crear, combinar y adaptar la plataforma a medida de sus requerimientos. Tiene funciones como las siguientes:

- Minimización de desarrollo
- Control máximo del rendimiento de la app. Mediante analíticas.
- Gestión fácil de todos los usuarios de la aplicación.
- Rapidez y agilidad.

### 4.4 Lenguajes

#### 4.4.1 UML

“Unified Modeling Language” o “Lenguaje Unificado de Modelado” es una notación centrada en objetos. Representa cada vista del sistema en diferentes diagramas que

constituyen la arquitectura global del sistema, como por ejemplo, los diagramas de secuencia que se han representado en el Anexo III – Análisis de requisitos.

#### 4.4.2 HTML

HTML es un lenguaje de programación muy sencillo que se utiliza para desarrollo web. HTML es la abreviatura de **H**yper **T**ext **M**arkup **L**anguage, es un lenguaje formado por etiquetas leídas por el navegador. La implementación de dichas etiquetas formará la estructura que verá el usuario por pantalla.

```
<div class="case-study-logo-container">
  <div id="main-hero" class="hero-body">
    <div class="container has-text-centered">
      <div class="columns is-vcentered">
        <div class="column is-6 is-offset-3 has-text-centered is-subheader-caption">
          <h1 class="title is-2">A/an - indefinite article</h1>
          <h2 class="subtitle">Compleat with 'a' or 'an'</h2>
        </div>
      </div>
    </div>
  </div>
</div>
```

Figura 8. Código HTML

#### 4.4.3 CSS

Cascading Styles Sheets, son las hojas de estilo que constituyen el diseño de cada uno de los elementos de HTML, con ello podemos dar estética a los componentes del sistema, dotando de colores, múltiples fuentes, formas, volumen, disposición... Consiguiendo que la interfaz de usuario sea más visual y atractiva, pero sin olvidarse de las buenas prácticas de usabilidad.

```
h1, h2, h3, h4, h5 {
  font-size: 100%;
  font-weight: normal
}
```

Figura 9. Código CSS

La última versión de CSS es la CSS3. La diferencia de una página Web sin CSS y otra con CSS es evidentemente notable y la podemos ver en un ejemplo.

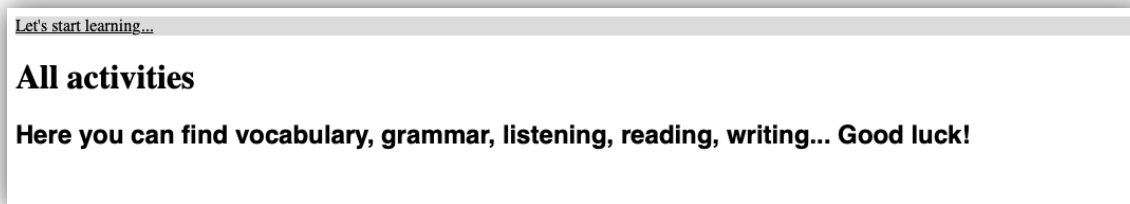


Figura 10. Página Web sin CSS

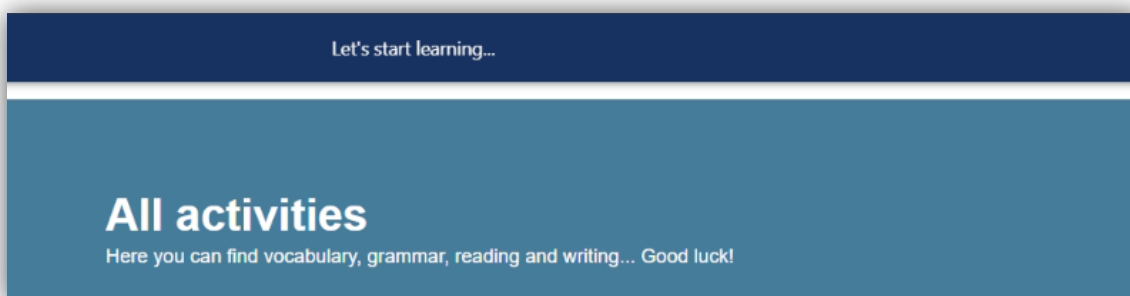


Figura 11. Página web con CSS

La figura 10 está libre de hojas de CSS, por ello los elementos como el título, imágenes o cajas de texto aparecen apiladas una encima de otra y sin ninguna estética. Al contrario de la Figura 11 que está adaptada con un documento de CSS para la redistribución y modificación de atributos de los elementos.

#### 4.4.5 JavaScript

JavaScript es el lenguaje de programación orientado a objetos que define un conjunto de instrucciones para generar un comportamiento dentro de la página web, puede ser desde una animación hasta la generación de un email.

En resumen, proporciona recursos interactivos en la página web, para dar una sensación de operabilidad y dinamismo. Permite incorporar elementos interactivos como relojes, contador de visitas, animaciones...

Un código simple de JavaScript es el de la Figura 12, que tiene como resultado en el momento de la carga de la página web un Pop-up con el literal introducido. En la figura 13 se ve el resultado de la ejecución del código javascript.

```

<!DOCTYPE html>
<html lang="es">
  <head></head>
  <body>
    <script type="text/javascript">
      alert("Bienvenido al Grado de Ingenieria informatica!!")
    </script>
  </body>
</html>

```

Figura 12. Código JavaScript

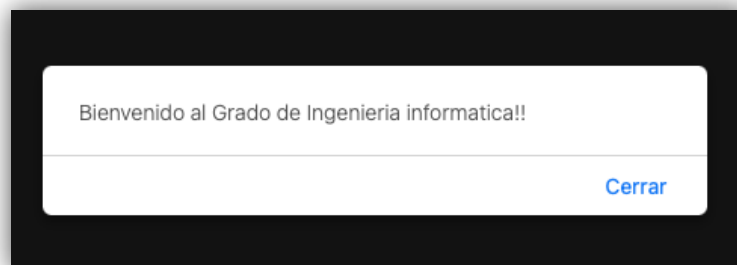


Figura 13. Ejecución código JavaScript

#### 4.4.6 JSON

El objeto de JavaScript (JavaScript Object Notation) se trata de un formato en tipo texto para intercambiar información. Resulta un formato sencillo de leer y escribir para los humanos y simple de interpretar y generar para las máquinas.

Es el formato que se ha usado en el almacenamiento y transmisión con la base de datos de Firebase por ejemplo.

### 4.5 Hosting y herramientas de compilación

#### 4.5.1 Node.js

Es un entorno de JavaScript, implementado en el backend, multiplataforma y con ventajas de rápido aprendizaje, alto rendimiento y soporte de alta concurrencia de usuarios con un solo servidor.

#### 4.5.2 Npm

Es un administrador de paquetes para los proyectos de Node.js disponibles. Su ventaja es la reutilización de múltiples códigos implementados por la comunidad.

### 4.6 Herramientas de control de versiones

#### 4.6.1 GitHub

GitHub es una herramienta utilizada en el control de versiones en el desarrollo de software. Tiene opción de utilizarse de forma colaborativa, y sirve para alojar las distintas versiones de los proyectos. Es una de las más populares y utilizadas en el mercado.

### 4.7 Herramientas CASE

#### 4.7.1 Visual Paradigm

Software para modelar diagramas en UML de una manera fácil e intuitiva, como si de un dibujo se tratara. Nos permite establecer relaciones entre los distintos componentes y clases, creando diversos diagramas para representar la arquitectura de un proyecto.

#### 4.7.2 Microsoft Project

Software que nos permite generar un flujo de trabajo desde el inicio del proyecto hasta su final. Permite generar las tareas necesarias para el desarrollo software con la duración de los días y los recursos disponibles, con todo ello generará un diagrama de Gantt para guiar y orientar al desarrollador en la construcción del software.

#### 4.7.3 EZEstimate

Esta herramienta nos predice de manera automática en función de la complejidad de actores, casos de usos, factores de complejidad técnica y de entorno, una estimación de horas/persona que durará el desarrollo del proyecto.

## 4.8 Herramientas de documentación

### 4.8.1 TypeDoc

Esta herramienta es o librería utilizada en proyectos software para dotar al código de una documentación “automatizada” o generada en el navegador de forma automática de una manera clara a través de comentarios en el propio código.

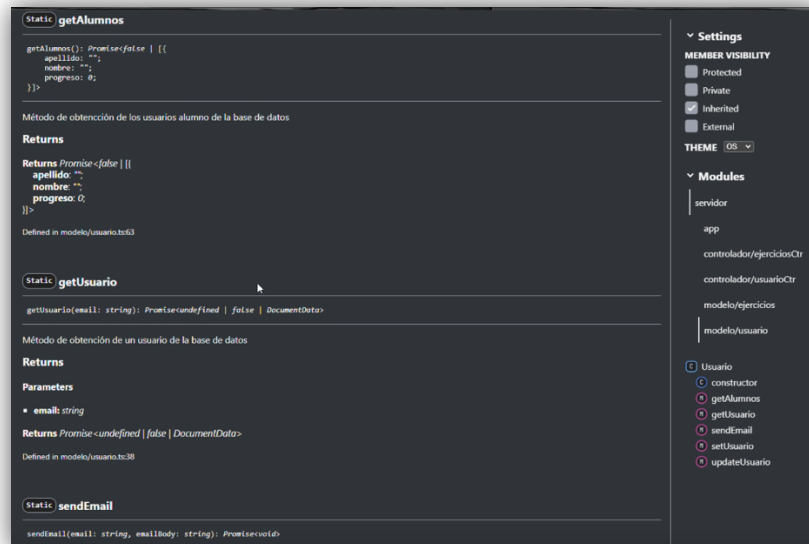


Figura 14. Ejemplo 9 TypeDoc

## 5 Aspectos relevantes del desarrollo

En este apartado explicaremos las partes más relevantes en el desarrollo del proyecto.

### 5.1 Marco de trabajo

La metodología que hemos seguido y como hemos hecho mención anteriormente es el Proceso Unificado.

El proceso Unificado es más que un simple proceso (Jacobson et al., 1999), es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas *software*, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

Dicho proceso dirigido por casos de uso y centrado en una arquitectura al ser iterativa e incremental.

iterativo e incremental ya que está compuesto de cuatro fases: Inicio, Elaboración, Construcción y Transición, a su vez divididas en iteraciones como resultado de la evolución incremental del proyecto.

Se dice que está dirigido por casos de uso ya que estos casos de uso documentan los requisitos del proyecto, por cada iteración se van nombrando a distintos casos de uso.

Las fases del proceso unificado son los que se denotan en la figura 15 de a continuación:

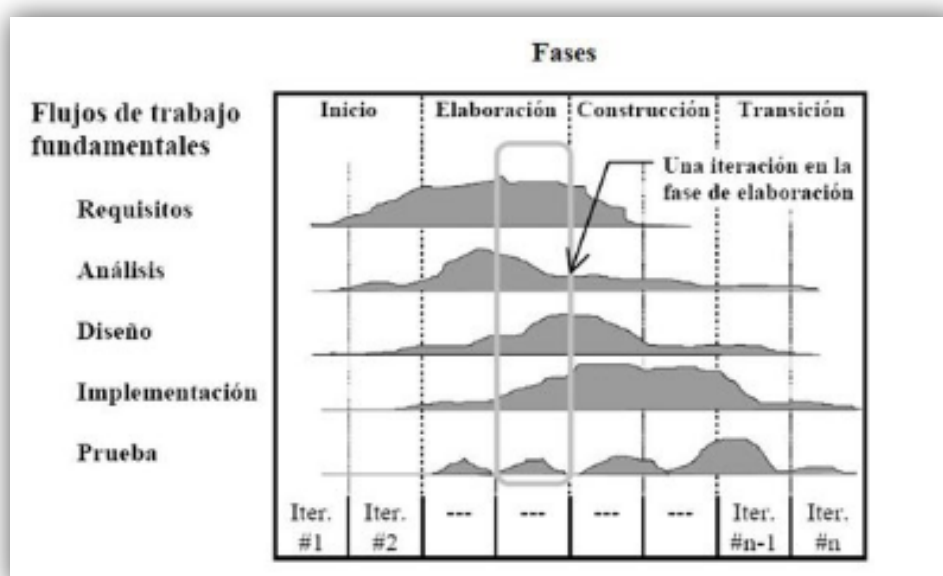


Figura 15. Proceso Unificado

1. Inicio: idealmente no se tiene gran conocimiento del proyecto, es una primera toma de contacto, en él se definen aspectos como plazos, presupuesto, alcance, viabilidad, y peticiones del usuario.
2. Elaboración: en esta segunda fase se refina la información del proyecto, se añaden requisitos y una estimación más exhausta.
3. Construcción: en esta tercera fase se lleva a cabo la propia implementación del proyecto.
4. Transición: en la última fase lo anterior debe estar acabado y listo para desplegarse sin mayor costo, también aquí recogemos test completos por parte de usuarios y/o cliente.

A lo largo del proyecto se llevan a cabo las distintas iteraciones dentro de cada fase, repitiéndose en cada una de ellas n veces según la importancia. Figura 16.

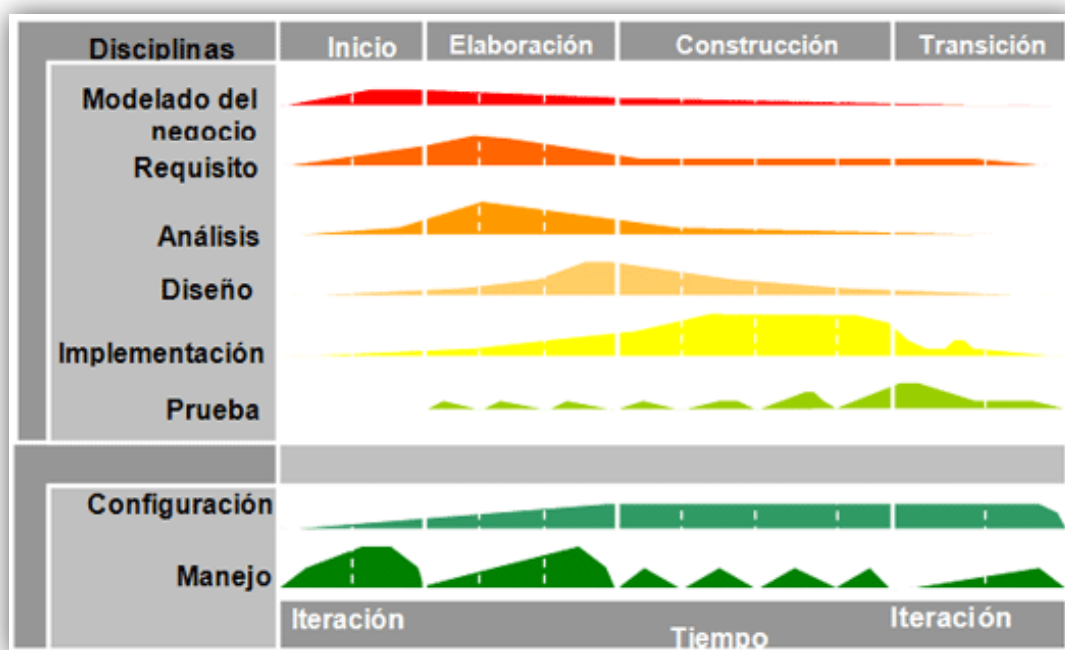


Figura 16. Iteraciones Proceso Unificado

1. **Modelado de negocio:** se hace un informe de la foto actual del proyecto, con su propio seguimiento.
2. **Requisitos:** se especifican los requisitos del sistema de forma detallada.
3. **Análisis:** se refinan los casos de uso y se establece su comportamiento.

4. **Diseño:** se construirá el diseño funcional y el comportamiento de los distintos elementos del sistema.
5. **Implementación:** se realizará la implementación del sistema.
6. **Pruebas:** se. Llevará a cabo las pruebas unitarias, funcionales y con el usuario del sistema.

## 5.2 Estimación de coste y esfuerzo

En la estimación de coste y esfuerzo se ha realizado la estimación del desarrollo del proyecto, utilizando una métrica de puntos casos de uso (UCP), a su vez, diversos factores de complejidad y relevancia que pesan en la predicción de la estimación, dichos factores están relacionados con conocimientos técnicos de programación y habilidades o capacidades relacionados con análisis o desarrollo del software, a su vez también se tiene en cuenta complejidad del entorno.

Esta estimación se ha desarrollado con la herramienta EZEstimate, utilizada en el Grado de Ingeniería Informática. Esta herramienta nos predice de manera automática en función de la complejidad de actores, casos de usos y los factores nombrados anteriormente una estimación de horas/persona que durará el desarrollo del proyecto.

Esta estimación se ha generado después de realizar la especificación de los requisitos, cuando ya se tienen claros los términos anteriores, el número de actores y su complejidad, al igual que los distintos casos de uso, dónde podemos saber en función de los pasos de secuencia normal que tiene su comportamiento la complejidad de este.

| Estimation Summary              |          |
|---------------------------------|----------|
| UAW                             | 9        |
| UUCW                            | 125      |
| UUPC = UAW + UUCW               | 134      |
| TFactor                         | 21       |
| EFactor                         | 28       |
| TCF = 0.6 + (.01*TFactor)       | 0.81     |
| EF = 1.4 + (-0.03*EFactor)      | 0.56     |
| UCP = UUPC*TCT*EF               | 60.7824  |
| <b>Total Effort@</b> 20 Hrs/UCP | 1215.648 |

Figura 17. Resultado de estimación de coste y esfuerzo

En la Figura 17 se puede observar los resultados de la estimación del esfuerzo.  
Para ver más información se puede consultar el Anexo I – Plan de Proyecto Software.

### 5.3 Planificación temporal

En este apartado veremos la tarea de planificación temporal que se llevó a cabo en el *Anexo I – Plan de proyecto*.

En cuanto a la planificación temporal se ha realizado en Microsoft Project, herramienta destinada para la temporalización de los proyectos, pudiendo determinar el tiempo de proyecto, las tareas de este y los recursos implicados. A su vez aporta gráficos e informes sobre este, siendo el más relevante el diagrama de Gantt, dónde muestra la secuencia de las tareas establecidas, nos remarca también las tareas críticas que hay durante el desarrollo.

Esta planificación sirve de guía durante toda la construcción del software, dando a conocer al desarrollador las tareas que debe realizar en cada momento y la duración del mismo, así como la necesidad de que no se retrasen en el caso de ser una tarea crítica.

El proyecto seguirá como se ha explicado anteriormente el flujo de trabajo propio del proceso Unificado, teniendo las cuatro fases como se muestra en la Figura 18.

| t | Nombre de tarea    | Duración | Comienzo     | Fin          | Predecesoras | re |
|---|--------------------|----------|--------------|--------------|--------------|----|
| ▶ | Inicio             | 41 días? | lun 01/08/22 | mar 30/08/22 |              |    |
| ▶ | Fin de Inicio      | 0 días   | mié 31/08/22 | mié 31/08/22 | 1FC+2 días   |    |
| ▶ | Elaboración        | 60 días  | mié 31/08/22 | mié 12/10/22 |              |    |
| ▶ | Fin de elaboración | 0 días   | jue 13/10/22 | jue 13/10/22 | 26FC+2 días  |    |
| ▶ | Construcción       | 95 días  | jue 13/10/22 | lun 19/12/22 |              |    |
| ▶ | Fin Construcción   | 0 días   | mar 20/12/22 | mar 20/12/22 | 70FC+2 días  |    |
| ▶ | Transición         | 27 días  | mié 21/12/22 | lun 09/01/23 | 114          |    |
| ▶ | Fin transición     | 0 días   | mar 10/01/23 | mar 10/01/23 | 115FC+2 días |    |

Figura 18. Proceso unificado mpp

En la Figura 18 se puede observar la planificación temporal en Microsoft Project.  
Para ver más información se puede consultar el Anexo I – Plan de Proyecto Software.

En las figuras siguientes mostraremos el diagrama de Gantt generado en las distintas fases del proyecto.

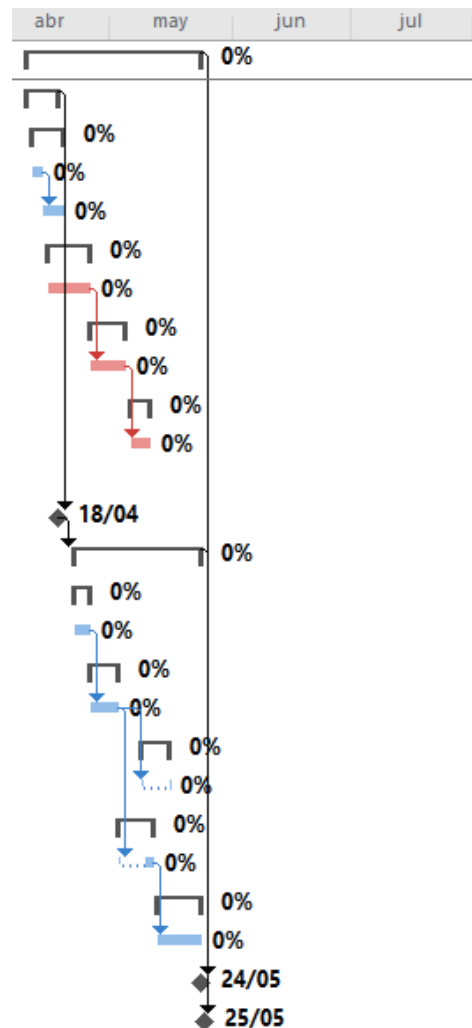


Figura 19. Diagrama de Gantt – Etapa de inicio

En la Figura 19 se puede observar la planificación temporal en Microsoft Project. Para ver más información se puede consultar el Anexo I – Plan de Proyecto Software.

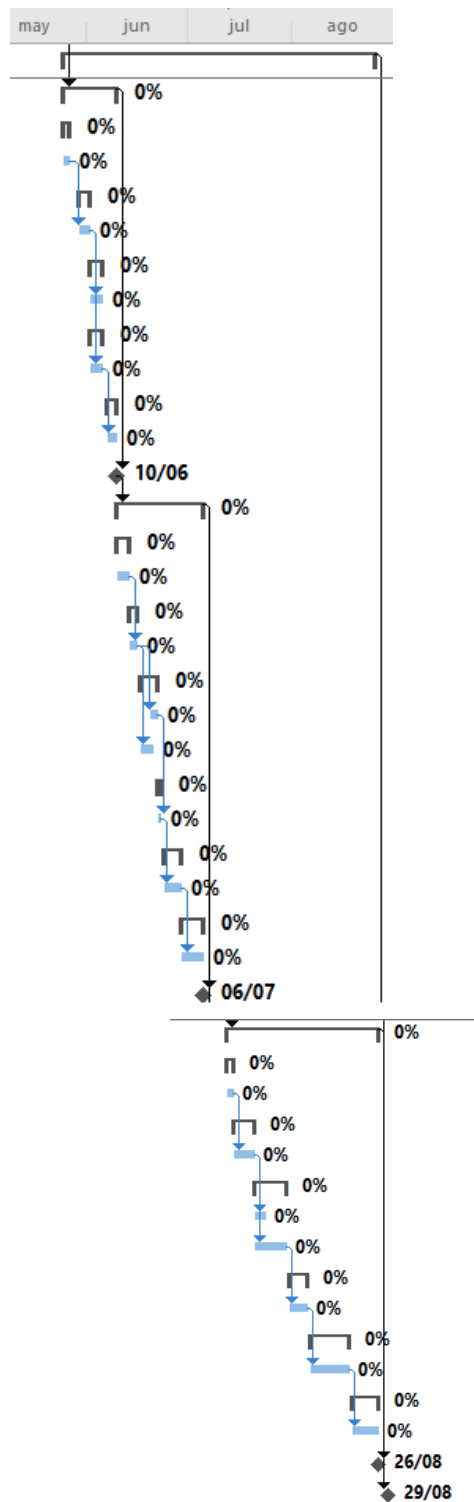


Figura 20. Diagrama de Gantt – Etapa de elaboración

En la Figura 20 se puede observar la planificación temporal en Microsoft Project. Para ver más información se puede consultar el Anexo I – Plan de Proyecto Software.

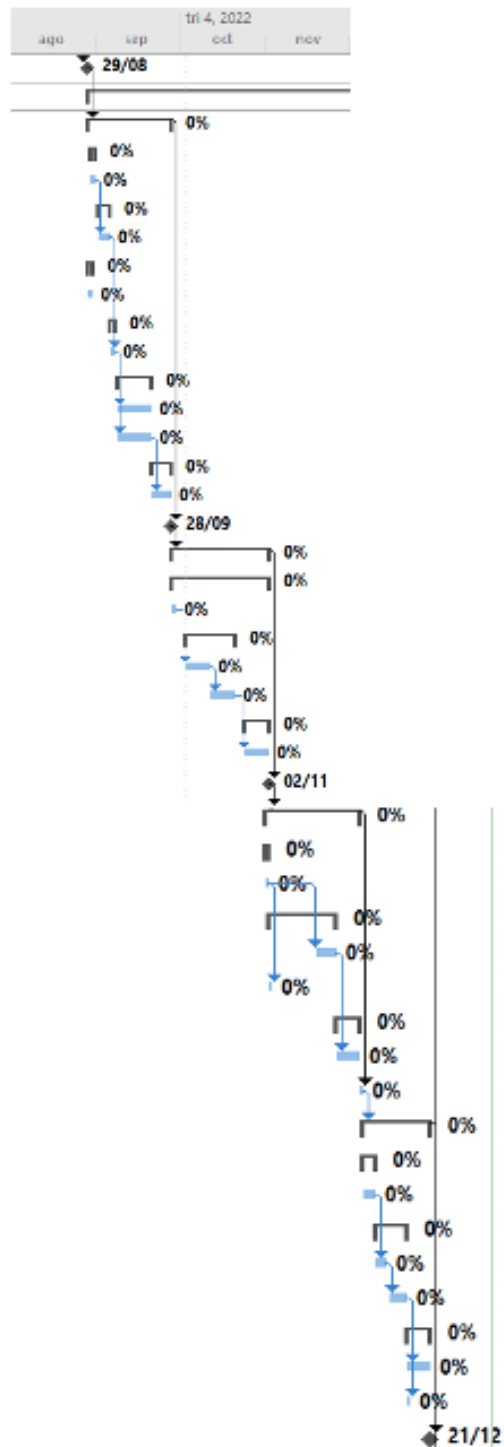


Figura 21. Diagrama de Gantt – Etapa de construcción

En la Figura 21 se puede observar la planificación temporal en Microsoft Project.  
Para ver más información se puede consultar el Anexo I – Plan de Proyecto Software.

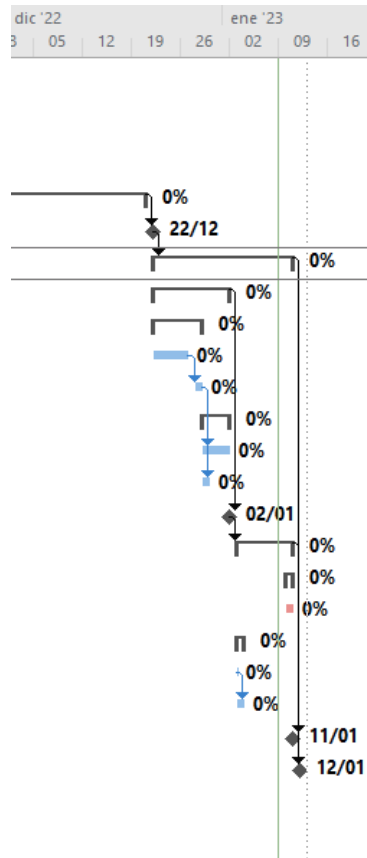


Figura 22. Diagrama de Gantt – Etapa de construcción

En la Figura 22 se puede observar la planificación temporal en Microsoft Project. Para ver más información se puede consultar el Anexo I – Plan de Proyecto Software.

## 5.4 Especificación de requisitos

En esta fase como su propio nombre nos dice, se llevará a cabo la especificación de los requisitos, participantes, objetivos, actores, casos de uso... *Anexo II – Especificación de requisitos.*

### 5.4.1 Participantes

El proyecto está compuesto por cuatro participantes, tres de ellos tutores y uno el alumno/desarrollador :

- Angel Cantuche Martin
- Sales Mendes, André Filipe.
- Villarrubia González, Gabriel

A continuación, se muestra una tabla de la especificación de participantes. Tabla 18.

|                                 |                       |
|---------------------------------|-----------------------|
| <b>Nombre del participante:</b> | Ángel Cantuche Martin |
| <b>Rol:</b>                     | Desarrollador         |
| <b>Desarrollador:</b>           | Sí                    |
| <b>Cliente:</b>                 | No                    |
| <b>Usuario:</b>                 | No                    |
| <b>Comentarios:</b>             | N/A                   |

Tabla 1. Autor 1

En la tabla 1 se puede observar la tabla de participante.

Para ver más información se puede consultar el Anexo II – Especificación de requisitos.

#### 5.4.2 Objetivos

Se especifican los objetivos que debe cumplir al final del proyecto. Los objetivos son los siguientes:

- Interacción del usuario con el sistema
- Presentación y visualización de los recursos del sistema
- Corrección de respuestas
- Muestra de las soluciones a los ejercicios
- Modificación de actividades

Se muestra a continuación una tabla de ejemplo sobre la especificación de los objetivos.

|                     |                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------|
| <b>OBJ-0001</b>     | <b>Interacción del usuario con el sistema</b>                                                |
| <b>Versión</b>      | 1.0                                                                                          |
| <b>Autores</b>      | Ángel Cantuche Martín                                                                        |
| <b>Fuentes</b>      | De Paz Santana, Juan Francisco<br>Sales Mendes, André Filipe<br>Villarubia González, Gabriel |
| <b>Descripción</b>  | El usuario podrá navegar por la página libremente, accediendo a los distintos recursos.      |
| <b>Subobjetivos</b> | N/A                                                                                          |
| <b>Prioridad</b>    | Alta                                                                                         |

Tabla 2. Objetivo 001

En la tabla 2 se puede observar la tabla de objetivo.

Para ver más información se puede consultar el Anexo II – Especificación de requisitos.

### 5.4.3 Actores

Este apartado muestra la especificación que se ha realizado de los actores del sistema, siendo estos los siguientes Figura 20:

- Usuario web Docente: en un marco educativo, como clases particulares o centros educativos. Es el usuario que forma a la demás gente.
- Usuario web: es el usuario sin rol docente que accede a la página web.
- Usuario web administrador: es el usuario que tiene plenos permisos en la página web.



Figura 23. Actores

En la Figura 23 se puede observar el diagrama de actores del sistema.

Para ver más información se puede consultar el Anexo II – Especificación de requisitos.

A continuación, se muestra una tabla de la especificación de los actores.

| ACT-0001    |  | Administrador                                                                                |
|-------------|--|----------------------------------------------------------------------------------------------|
| Versión     |  | 1.0                                                                                          |
| Autores     |  | Ángel Cantuche Martín                                                                        |
| Fuentes     |  | De Paz Santana, Juan Francisco<br>Sales Mendes, André Filipe<br>Villarubia González, Gabriel |
| Descripción |  | Es el usuario que tiene plenos permisos en la página web.                                    |
| Comentarios |  | N/A                                                                                          |

Tabla 3. Actor 0001

En la tabla 3 se puede observar la tabla de actor en el sistema.  
Para ver más información se puede consultar el Anexo II – Especificación de requisitos.

#### 5.4.4 Diagrama de paquetes

Se ha distribuido en paquetes para una mayor claridad de requisitos y casos de uso.  
Figura 22.

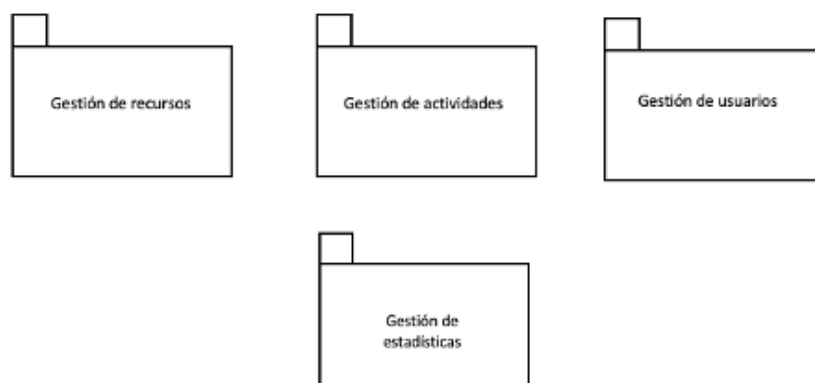


Figura 24. Diagrama de paquetes.

En la Figura 24 se puede observar el diagrama de paquetes del sistema.  
Para ver más información se puede consultar el Anexo II – Especificación de requisitos

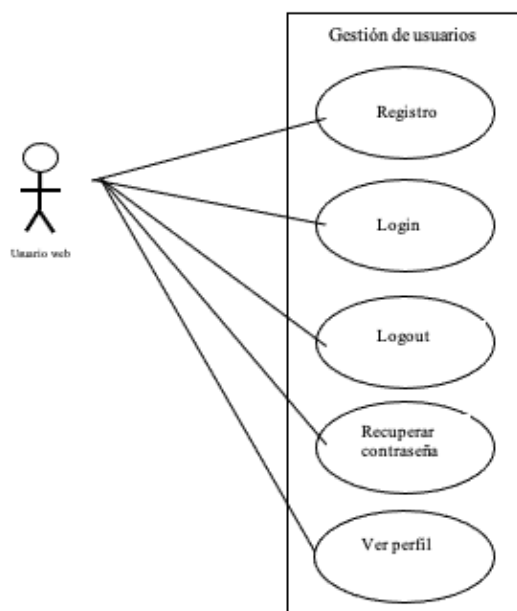


Figura 25. Diagrama paquete gestión de usuarios.

En la Figura 25 se puede observar el diagrama del paquete de gestión de usuarios.  
Para ver más información se puede consultar el Anexo II – Especificación de requisitos.

Los requisitos funcionales definen el comportamiento del sistema en los distintos casos.  
En la siguiente tabla veremos un caso de uso y el comportamiento que debe tener el sistema ante las acciones del usuario. Tabla 4.

| UC-0001          |                                                                                                                          | Login página web                                                                                                     |  |
|------------------|--------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|--|
| Versión          | 1.0                                                                                                                      |                                                                                                                      |  |
| Autores          | Ángel Cantuche Martín                                                                                                    |                                                                                                                      |  |
| Fuentes          | De Paz Santana, Juan Francisco<br>Sales Mendes, André Filipe<br>Villarrubia González, Gabriel                            |                                                                                                                      |  |
| Dependencias     | OBJ-0001 – Interacción del usuario con el sistema<br>OBJ-0002 - Presentación y visualización de los recursos del sistema |                                                                                                                      |  |
| Descripción      | El sistema deberá mostrar la información y trabajar como se muestra en la secuencia normal                               |                                                                                                                      |  |
| Precondición     | -                                                                                                                        |                                                                                                                      |  |
| Secuencia normal | PASO                                                                                                                     | ACCIÓN                                                                                                               |  |
|                  | 1                                                                                                                        | El actor Usuario web ACT-0003 o el actor Usuario Web Docente ACT-0002 solicita hacer login en el sistema             |  |
|                  | 2                                                                                                                        | El sistema solicita al usuario los datos de autenticación.                                                           |  |
|                  | 3                                                                                                                        | El actor Usuario web ACT-0003 o el actor Usuario Web Docente ACT-0002 introduce los datos solicitados por el sistema |  |
|                  | 4                                                                                                                        | El sistema autentica al usuario si los datos son correctos                                                           |  |
| Postcondición    |                                                                                                                          |                                                                                                                      |  |
| Excepciones      | El usuario no se encuentra registrado en la base de datos del sistema.                                                   |                                                                                                                      |  |
| Estabilidad      | Alta                                                                                                                     |                                                                                                                      |  |

Tabla 4. UC-0001

En la tabla 4 se puede observar la tabla de caso de uso.

Para ver más información se puede consultar el Anexo II – Especificación de requisitos.

### 5.5.6 Requisitos de información

En estos requisitos de información indicamos los datos que el sistema debe almacenar.

- Información de usuarios
- Información sobre actividades
- Información de progreso
- Información sobre categorías

| IRQ-0001 Información de usuarios |                                                                                                                     |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <b>Versión</b>                   | 1.0                                                                                                                 |
| <b>Autores</b>                   | Ángel Cantuche Martín                                                                                               |
| <b>Fuentes</b>                   | De Paz Santana, Juan Francisco<br>Sales Mendes, André Filipe<br>Villarubia González, Gabriel                        |
| <b>Dependencias</b>              | Ninguno                                                                                                             |
| <b>Descripción</b>               | El sistema deberá almacenar información correspondiente a los usuarios del sistema.                                 |
| <b>Datos específicos</b>         | <ul style="list-style-type: none"><li>- Identificador</li><li>- Nombre</li><li>- Apellido</li><li>- Email</li></ul> |
| <b>Importancia</b>               | Importante                                                                                                          |
| <b>Urgencia</b>                  | Inmediatamente                                                                                                      |
| <b>Estado</b>                    | Validado                                                                                                            |
| <b>Estabilidad</b>               | Alta                                                                                                                |
| <b>Comentarios</b>               | N/A                                                                                                                 |

Tabla 5. IRQ 0001

En la tabla 5 se puede observar la tabla de requisitos de información.

Para ver más información se puede consultar el Anexo II – Especificación de requisitos.

### 5.5.7 Requisitos no funcionales

Aquí se va a tratar los requisitos no funcionales, que son aquellos requisitos que especifican una norma o restricción ante el sistema.

Los requisitos funcionales son los siguientes:

- Escalabilidad
- Concurrencia
- Usabilidad
- Eficiencia
- Disponibilidad

A continuación, se muestra la tabla de los requisitos no funcionales:

| NFR-0001    |  | Escalabilidad                                                                                |  |
|-------------|--|----------------------------------------------------------------------------------------------|--|
| Versión     |  | 1.0                                                                                          |  |
| Autores     |  | Ángel Cantuche Martín                                                                        |  |
| Fuentes     |  | De Paz Santana, Juan Francisco<br>Sales Mendes, André Filipe<br>Villarubia González, Gabriel |  |
| Descripción |  | El sistema deberá crecer a medida que<br>pasa el tiempo.                                     |  |
| Prioridad   |  | Alta                                                                                         |  |

Tabla 6. NFR-0001

En la Figura 6 se puede observar la tabla de requisitos de información.

Para ver más información se puede consultar el Anexo II – Especificación de requisitos.

## 5.5 Análisis de requisitos

En este apartado veremos el análisis de los requisitos anteriores que se ha llevado a cabo. Todo ello y para más información está recogido en el *Anexo III – Análisis de requisitos*.

### 5.5.1 Modelo de dominio

El modelo de dominio representado con un diagrama de clases muestra los objetos que hay en el sistema y la relación entre ellos.

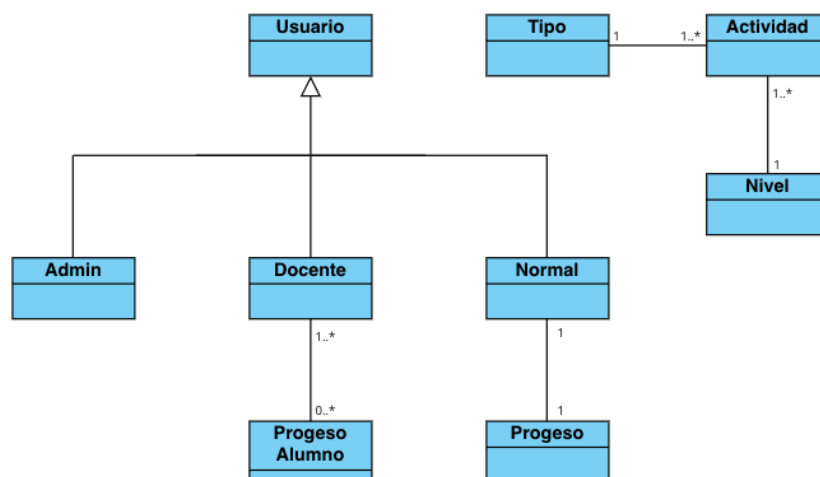


Figura 26. Modelo de dominio

En la Figura 26 se puede observar el modelo de dominio.

Para ver más información se puede consultar el Anexo III – Análisis de requisitos.

- **Actividad**: representa el conjunto de actividades de la página Web.
- **Tipo**: representa la categoría de la actividad, gramática, vocabulario, Reading. Una categoría puede tener muchas actividades, pero una actividad solo pertenecerá a su categoría correspondiente.
- **Nivel**: representa el nivel de dificultad que tiene la actividad. Una actividad puede tener más de un nivel y un nivel representa a una o más actividades.
- **Usuario**: representa los usuarios del sistema, separando los distintos roles.
- **Progreso**: representa el progreso obtenido hasta la fecha de los usuarios, un usuario tiene un progreso, y el mismo progreso solo pertenece al usuario.
- **Progreso Alumno**: representa el progreso de un conjunto de alumnos de forma aislada, para que el Docente pueda tener constancia del progreso de cada alumno.

### 5.5.2 Análisis de requisitos

En la figura 27, mostraremos una especificación más exhaustiva de un caso de uso, la representación será mediante diagrama de secuencia.

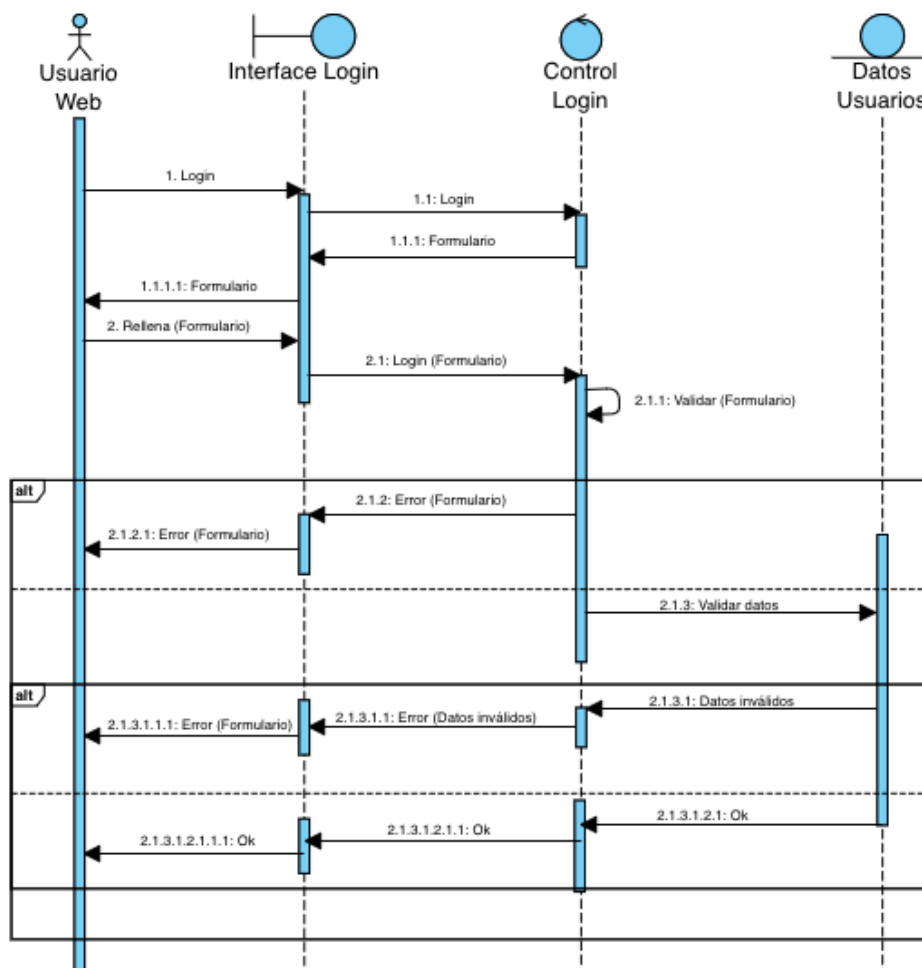


Figura 27. Análisis de requisito – diagrama de secuencia

En la Figura 27 se puede observar un diagrama de secuencia.

Para ver más información se puede consultar el Anexo III – Análisis de requisitos.

### 5.6.3 Clases de análisis

En el apartado de clases de análisis se han realizado diagramas de comunicación que relacionan las clases del punto anterior. En la Figura 28 se muestra un ejemplo de diagrama de comunicación, haciendo referencia al paquete de gestión de recursos.

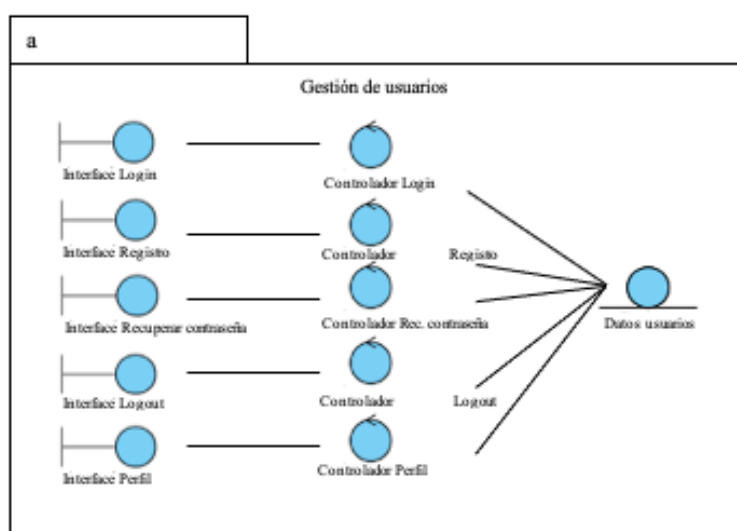


Figura 28. Clases de análisis - diagrama de comunicación

En la Figura 28 se puede observar un diagrama de comunicación.

Para ver más información se puede consultar el Anexo III – Análisis de requisitos.

## 5.6 Diseño del sistema

### 5.6.1 Patrones arquitectónicos

Patrón arquitectónico MVC, Modelo-Vista-Controlador.

Este patrón está clasificado como sistema interactivo, es decir, ayudan a organizar y estructurar el sistema con interacción hombre-máquina. Se encuentra en frameworks de aplicación para software con interfaces gráficas.

Se utiliza cuando hay interfaces de usuario que pueden cambiar al extender la funcionalidad del sistema o una modificación del propio entorno gráfico, es decir, con ello se construye un sistema flexible.

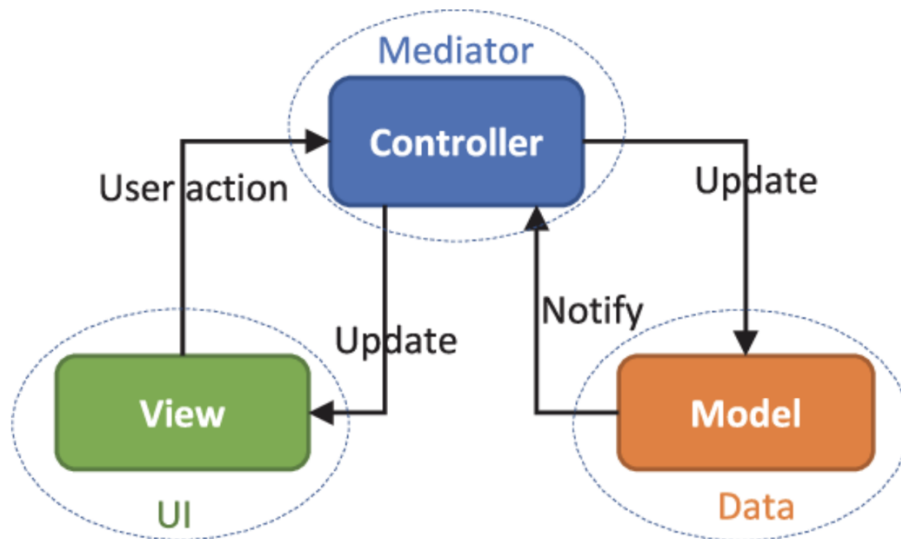


Figura 29. MVC

## 5.6.2 Subsistema del diseño

### 5.6.2.1 Subsistema de componentes

En este apartado veremos los componentes agrupados según su lógica en la aplicación y funcionalidad dentro del propio sistema.

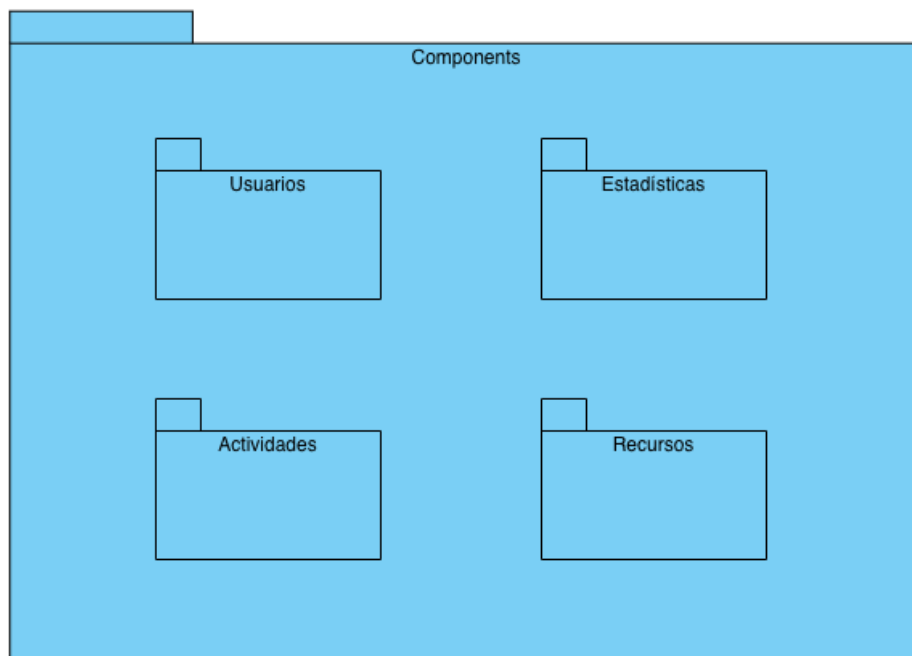


Figura 30. Diagrama Components

En la Figura 30 se puede observar un diagrama de comunicación.

Para ver más información se puede consultar el Anexo IV – Diseño del sistema.

### 5.6.2.2 Realización de casos de uso

Refinamiento de diagramas de secuencia, paquete Gestión de recursos.

Caso de uso: Recurso teórico

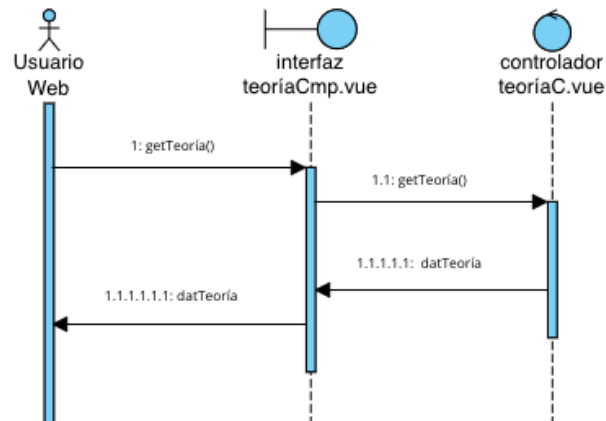


Figura 31. Diagrama de secuencia teoría

En la Figura 31 se puede observar un diagrama de comunicación.

Para ver más información se puede consultar el Anexo IV – Diseño del sistema.

### 5.6.2.3 Descripción de vista arquitectónica

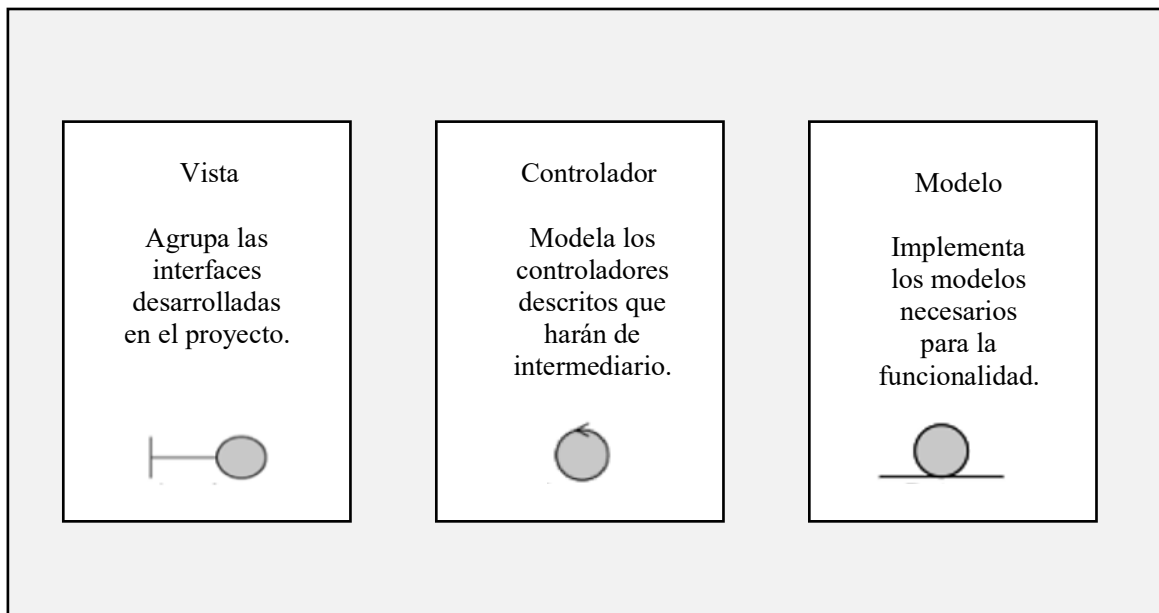


Figura 32. Descripción de vista arquitectónica

#### 5.6.2.4 Modelo de despliegue

El modelo de despliegue representa la ejecución del sistema, incluye los distintos nodos así como las capas intermedias.

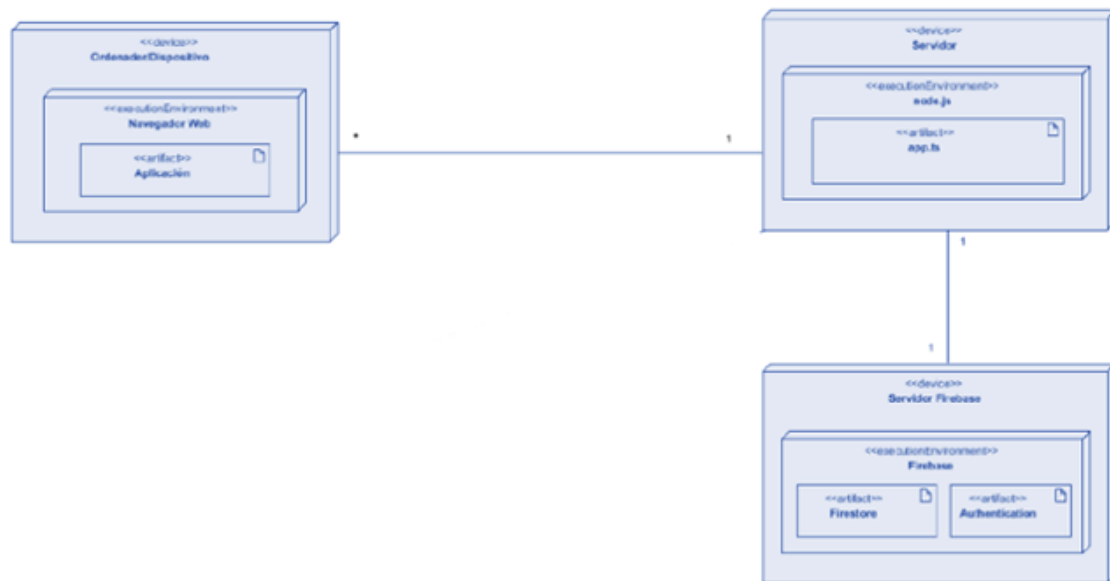


Figura 33. Descripción de vista arquitectónica

En la Figura 33 se puede observar un diagrama de comunicación.

Para ver más información se puede consultar el Anexo IV – Diseño del sistema.

## 6. Implementación

En esta fase es donde se implementa o construye todo el sistema, es una fase de gran peso dentro del proyecto debido a que ocupa gran parte de la estimación.

En esta fase se han puesto en marcha tecnologías nombradas en el apartado de Técnicas y herramientas anteriormente.

En esta fase es donde se ha implementado los códigos fuente dándole contenido a la aplicación, lógica de negocio y técnicas visuales de css para atribuir estética la cual sea atractiva para el usuario. Después de acabar con esta parte se ha completado la página Web con interacciones para que el usuario pueda navegar por el sistema y funcionalidad que dote a la página Web de un dinamismo propio de la navegación actual.

La implementación se ha diferenciado en dos notables partes:

- Backend: en esta parte se ha implementado todo el backend del sistema, representa la base de datos donde se almacenan los datos, es la parte transparente al usuario. Se ha utilizado la tecnología Node.js para generar los distintos archivos necesarios para la funcionalidad de los distintos recursos de la aplicación. Desde el frontend generaremos las peticiones node para la transmisión de la información requerida en cada caso.
- Frontend: en esta parte se ha implementado todo el software del sistema de parte del cliente. Desarrollamos tanto las interfaces que se presentan al usuario, como funcionalidades de comunicación con distintos componentes. Además, en esta parte también se genera la conexión con el servicio Firebase para la gestión general de usuarios.

## 7. Pruebas

A lo largo del desarrollo del proyecto se han ido realizando pruebas de menos a más en las distintas iteraciones de construcción.

Se han llevado a cabo pruebas unitarias de códigos para ver como era su visualización en las páginas y pruebas css para validar la estética en la página. Estos test se han realizado cada poco en la parte del código ya que en una página Web podemos ver al instante los efectos que genera un cambio o una adicción dentro del código.

Además, también se han ido realizando pruebas en cuanto a hipervínculos dentro de la propia página e interacción con otras ventanas.

Al finalizar su codificación se ponen en marcha pruebas con distintos usuarios que han estado al margen de la codificación, puesto que ellas no saben cómo funciona y es ahí dónde podemos estudiar la usabilidad y manejo de la página Web.

## 8. Aplicación

La funcionalidad de la aplicación se puede organizar en los siguientes puntos principales:

- Vista principal
- Login/Registro a la aplicación
- Perfil
- Recursos
- Listado de actividades
- Realización de actividades
- Contactar
- Cerrar sesión

A continuación, se van a describir de forma breve las principales funcionalidades. Una especificación completa y detallada de la funcionalidad ofrecida por la aplicación puede encontrarse en el Anexo V - Manual de usuario.

### 8.1 Vista principal

Cuando el usuario acceda a la homePage el sistema le mostrará la siguiente pantalla.

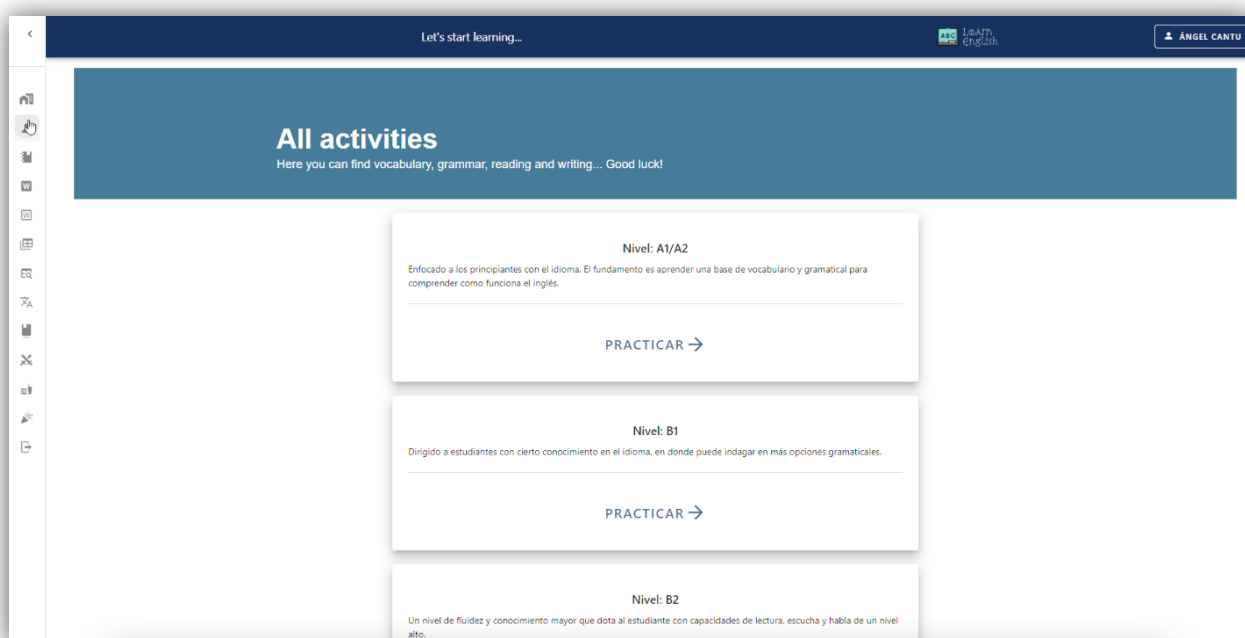


Figura 34. HomePage

## 8.2 Acceso a la aplicación

Al acceder a la aplicación aparecerá una primera pantalla donde se encontrarán dos opciones, una primera por defecto que será la opción de Login con un formulario de email y contraseña, si el usuario esta dado de alta en la aplicación podrá acceder a ella rellenando el formulario.

Si por el contrario no está registrado en el sistema deberá entrar en la opción de registro y rellenar ese formulario, si el sistema lo valida satisfactoriamente el usuario ya podrá acceder al sistema.

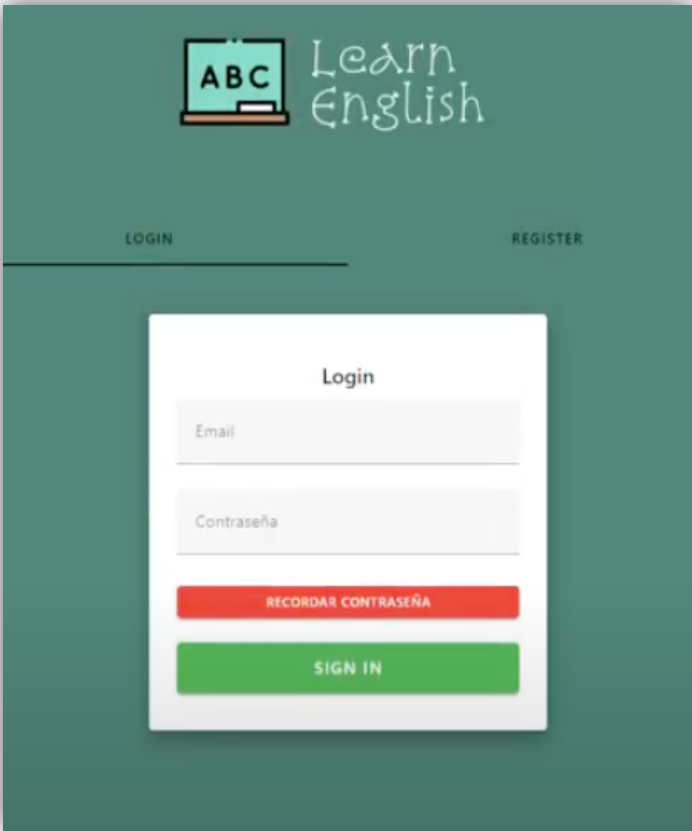
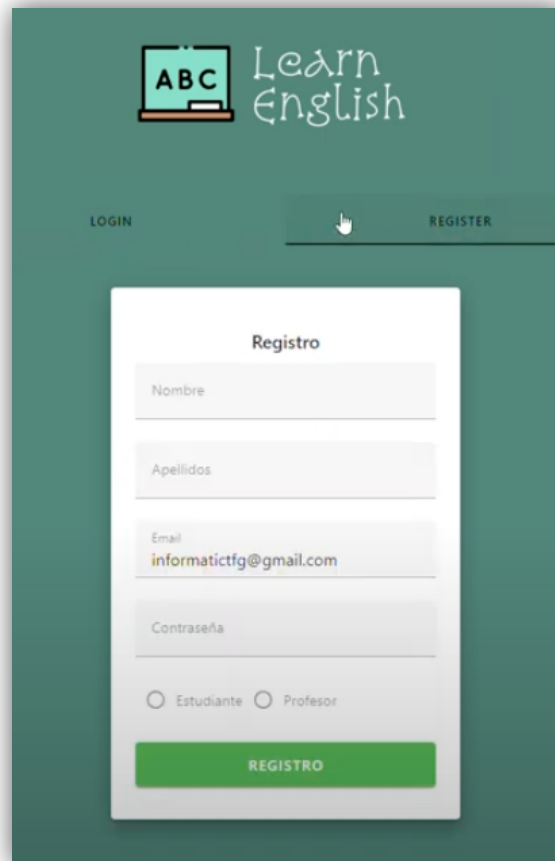


Figura 35. Login



The image shows a web interface for 'Learn English'. At the top, there is a logo with 'ABC' on a book and the text 'Learn English'. Below the logo, there are two buttons: 'LOGIN' and 'REGISTER'. The 'REGISTER' button is highlighted with a mouse cursor. Below these buttons is a registration form titled 'Registro'. The form contains the following fields: 'Nombre', 'Apellidos', 'Email' (with the value 'informatictf@gmail.com'), and 'Contraseña'. Below the password field are two radio buttons: 'Estudiante' and 'Profesor'. At the bottom of the form is a green button labeled 'REGISTRO'.

Figura 36. Registro

### 8.3 Perfil del usuario

Cuando el usuario acceda al sistema, este le mostrará su homePage, para ver su perfil el usuario deberá pulsar en el icono de usuario con su nombre de la esquina superior a la derecha. El sistema le mostrará su perfil y progreso, de esta manera también podrá editar el perfil.



Figura 37. Ver perfil

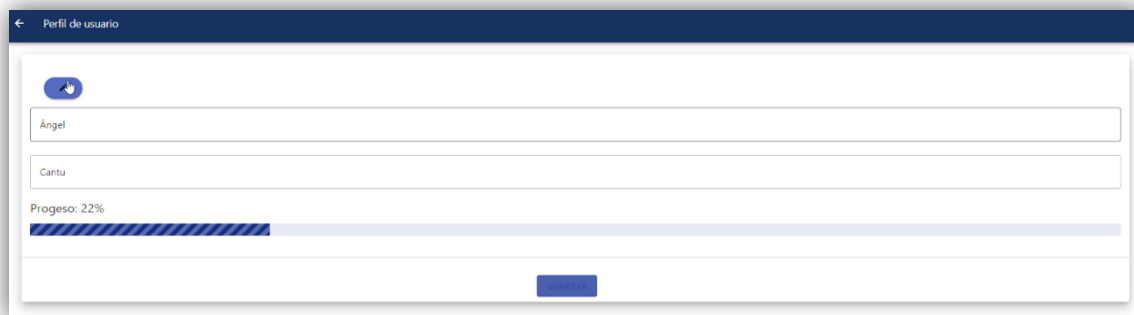


Figura 38. Modificar perfil

## 8.6 Recursos

La aplicación tiene información extra en el menú desplegable de la izquierda, en ese apartado el usuario podrá encontrar teoría extra, vocabulario, información sobre recetas, festividades y otros datos importantes.

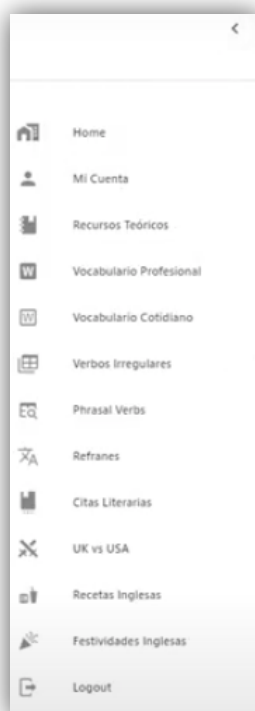


Figura 39. Menú desplegable

En la figura 40 de puede apreciar el recurso sobre verbos irregulares que proporciona la aplicación al usuario.

| Verbos Irregulares |             |                 |             |
|--------------------|-------------|-----------------|-------------|
| Infinitive         | Simple Past | Past Participle | Castellano  |
| arise              | arose       | arisen          | surgir      |
| be                 | was / were  | been            | ser         |
| beat               | beat        | beaten          | golpear     |
| become             | became      | become          | convertirse |
| begin              | began       | begun           | comenzar    |
| bet                | bet/betted  | bet/betted      | apostar     |
| bite               | bit         | bitten          | morder      |
| bleed              | bled        | bled            | sangrar     |
| blow               | blew        | blown           | soplar      |
| break              | break       | broken          | romper      |

Figura 40. Verbos irregulares

En las figuras 41-42 podemos ver el recurso sobre recetas y festividades inglesas que contiene el sistema para dar información adicional sobre tradiciones inglesas la aplicación al usuario.

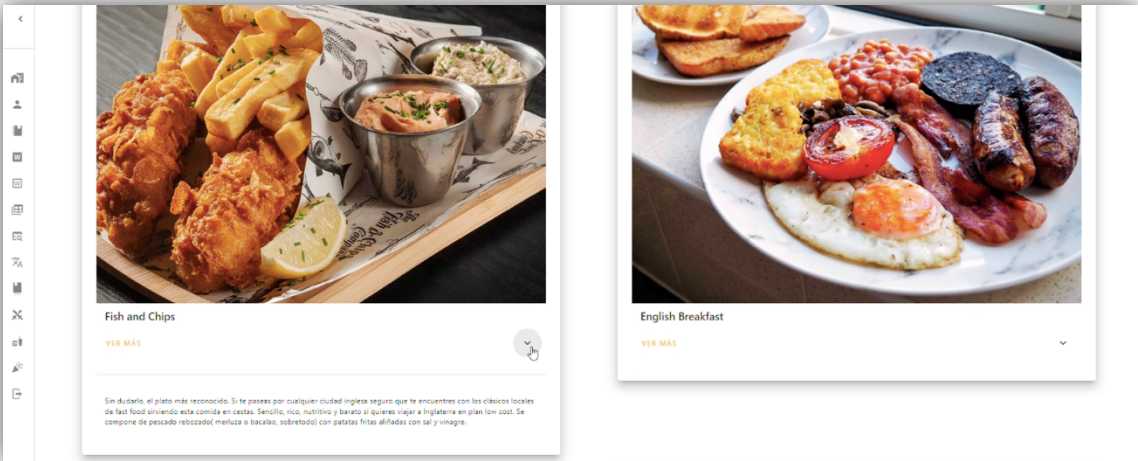


Figura 41. Recetas inglesas

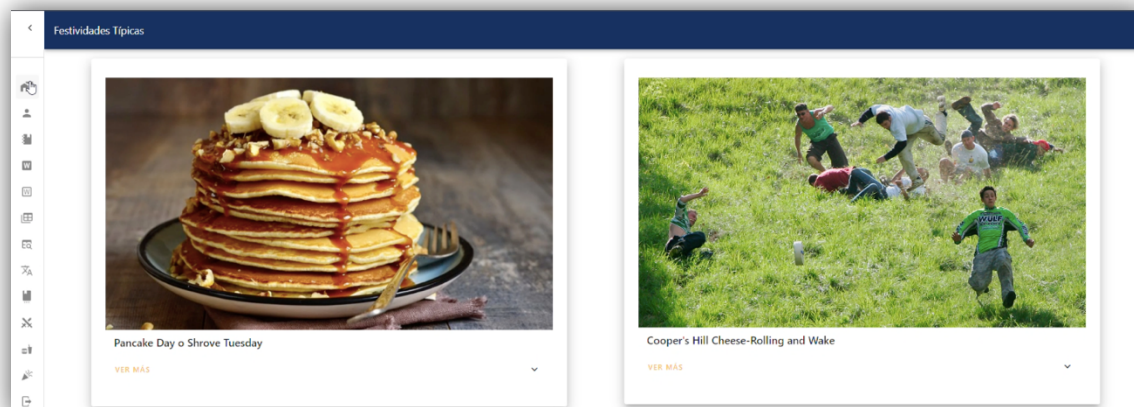


Figura 42. Recurso festividades inglesas

## 8.5 Listado de actividades / realización de actividades

Cuando el usuario quiera acceder a alguna actividad deberá pulsar en el nivel deseado de la homePage, posteriormente aparecerán un grupo de actividades como se ve en la figura 44, el usuario seleccionará el ejercicio que desee. El sistema le mostrará el ejercicio. El usuario deberá rellenar los huecos vacíos con sus respuestas y cuando este termine pulsará en corregir respuestas para que el sistema le muestre su puntuación, como se describe en la figura 45. En la figura 46 vemos como le aparecerá la opción de mostrar soluciones para que la aplicación muestre automáticamente las respuestas correctas del ejercicio.

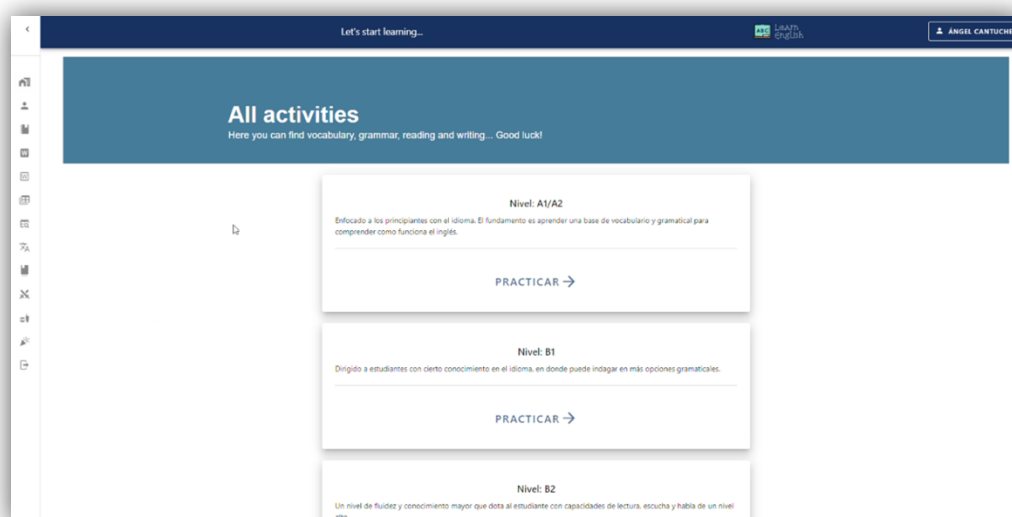


Figura 43. Niveles de inglés



Figura 44. Listado de actividades

|                                  |      |
|----------------------------------|------|
| 1. I have 1 car                  | 1.a  |
| 2. She is 2 engineer             | 2.an |
| 3. We need 3 big house           | 3.a  |
| 4. They are buying 4 apple       | 4.an |
| 5. Hello, we are in 5 university | 5.a  |
| 6. She will be here in 6 hour    | 6.an |
| 7. This is 7 hotel               | 7.   |
| 8. I want 8 hamburguer           | 8.   |
| 9. You will need 9 umbrella      | 9.   |
| 10. He is 10 nice guy            | 10.  |

**CORREGIR**

Figura 45. Corregir actividades

|                                  |       |
|----------------------------------|-------|
| 1. I have 1 car                  | 1. a  |
| 2. She is 2 engineer             | 2. an |
| 3. We need 3 big house           | 3. a  |
| 4. They are buying 4 apple       | 4. an |
| 5. Hello, we are in 5 university | 5. a  |
| 6. She will be here in 6 hour    | 6. an |
| 7. This is 7 hotel               | 7. a  |
| 8. I want 8 hamburguer           | 8. a  |
| 9. You will need 9 umbrella      | 9. an |
| 10. He is 10 nice guy            | 10. a |

**MOSTRAR RESPUESTAS**

Figura 46. Mostrar soluciones

## 8.6 Contacto

Cuando el usuario quiera ponerse en contacto con el administrador de la página deberá rellenar el formulario que aparece cuando pulsa en el icono de sobre del footer.

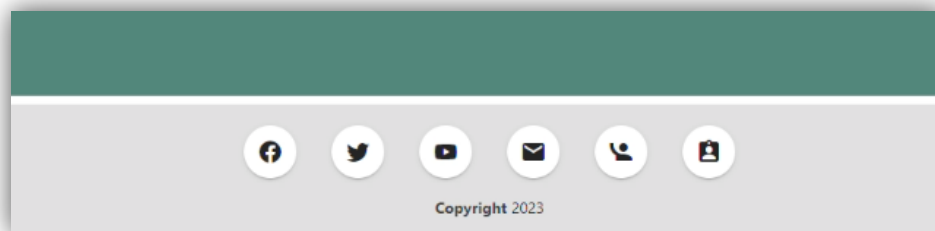
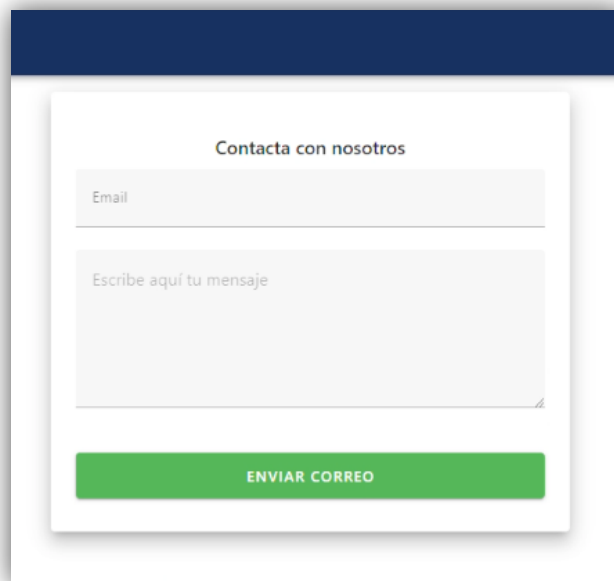


Figura 47. Icono contactar



Formulario de contacto con el título "Contacta con nosotros". El formulario contiene un campo de entrada para el correo electrónico etiquetado como "Email", un área de texto grande para el mensaje etiquetada como "Escribe aquí tu mensaje", y un botón verde que dice "ENVIAR CORREO".

Figura 48. Formulario contactar

## 8.6 Cerrar sesión

Cuando el usuario logueado desee cerrar sesión de la aplicación tendrá que desplegar el menu izquierdo de la pantalla de inicio y navegar hasta la última opción dónde pone logout, cuando el usuario pulse el sistema cerrará su sesión en el sistema. Figura 49.

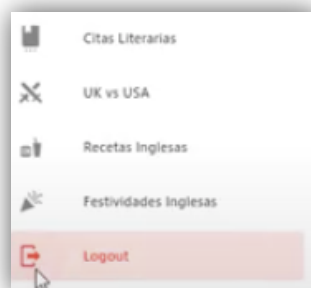


Figura 49. Logout

## 9 Conclusiones y líneas futuras

En este apartado veremos las conclusiones que emergen de este proyecto una vez terminado, y las posibles líneas de trabajo futuras para seguir con distintos evolutivos en la mejora de este.

### 9.1 Conclusiones

A continuación, terminado el proyecto podemos analizar los objetivos que se han superado, el balance y motivación de estos.

Después de la realización de este proyecto cabe destacar la carga que tiene un proyecto con sus correspondientes tareas, en el caso de no haber un equipo para su realización y para que apoye en las distintas fases.

Se nota también el uso de distintas tecnologías para su realización, se ha utilizado frameworks que sea más rápido su aprendizaje y mantenibilidad. También con el apoyo de la numerosa documentación que tienen todas estas, pudiendo corregir errores y solucionar dudas de una manera fácil y rápida.

A continuación, se enumerarán los objetivos del proyecto y se dará una breve explicación de la resolución de los mismos.

El objetivo primero era *la interacción del usuario con el sistema*, se ha logrado implementar una aplicación amigable e intuitiva para que el usuario final no tenga ningún problema de usabilidad, no requiere ningún entrenamiento previo, debido a lo anterior y a la familiarización que hay en la actualidad con las tecnologías.

El objetivo segundo trataba de *la presentación y visualización de los recursos*, se ha logrado desarrollar el sistema para que el usuario tenga disponibilidad inmediata de todos los recursos de la aplicación, poder visualizarlos de manera clara y sin ambigüedad.

El objetivo tercero estaba centrado en *la corrección de las respuestas a las actividades*, se ha logrado que el sistema automáticamente corrija las respuestas del usuario final ante los ejercicios del sistema, mostrando la puntuación que ha obtenido en él.

El objetivo cuarto, *muestra de las soluciones del ejercicio*, se ha logrado implementando una opción para que el usuario final pueda obtener las respuestas correctas de las actividades

El objeto quinto, *modificación de actividades*, se ha logrado gracias a un recurso del sistema que nos permite abrir un canal de comunicación para poder añadir, borrar o modificar las actividades del sistema en cualquier momento.

Por último, gracias a este proyecto he tenido la oportunidad de aplicar las técnicas que se han estudiado en el Grado de Ingeniería Informática, desde seguir una metodología ágil hasta tener conocimientos de cliente-servidor y diseño web desde el principio del proyecto hasta su final.

## 9.2 Líneas de trabajo futuras

Un punto importante para mejorar en el futuro sería la mejora o adición de nuevas funcionalidades en la página Web, como así mantenibilidad de este.

Creación de nuevos ejercicios a demanda de la oferta que hay en el mercado guiando al usuario como ahora a través de distintos ejercicios para la mejora del conocimiento de un idioma.

Así como el estudio estadístico de los usuarios por la página Web, bien número de ellos que acceden frente a los que acceden y además hacen uso de las distintas actividades, a su vez llevar un recuento de cuáles son las funcionalidades más visitadas por los usuarios.

## 10 Bibliografía

- McConnell, S. - “Desarrollo y gestión de proyectos informáticos”. Mc Graw Hill, 1997.
- Pressman, R. S. - “Ingeniería del wSoftware: Un Enfoque Práctico”. 7a Edición. McGraw-Hill. 2010.
- Martin Fowler, Knedall Scoot - “UML Gota a Gota”. Prentice Hall, 1999
- Visual paradigm, accedido en 2022
  - <https://www.visual-paradigm.com>
- Teoría sobre páginas web, accedido en 2022
  - <https://sites.google.com/site/pajinadetaller2/home/capitulo-i-introduccion/parte-2>
- Documentación Visual Studio Code, accedido en 2022
  - <https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- Vue.js, accedido en 2022
  - <https://lenguajejs.com/vuejs/introduccion/que-es-vue/>
- Documentación UML, accedido en 2022
  - [https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=688:ique-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46&Itemid=163](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:ique-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46&Itemid=163)
- Documentación Css, accedido en 2022
  - <https://developer.mozilla.org/es/docs/Web/CSS>
- Documentación HTML, accedido en 2022
  - <https://developer.mozilla.org/es/docs/Web/HTML>
- Documentación JavaScript, accedido en 2022
  - <https://developer.mozilla.org/es/docs/Web/JavaScript>
- TypeDoc, accedido en 2022
  - <https://typedoc.org/>
- Documentación Node.js, accedido en 2022
  - <https://openwebinars.net/blog/que-es-nodejs/>
- Blog de páginas Web responsives, accedido en 2022
  - <https://kinsta.com/es/blog/diseno-de-paginas-web-sensibles/>
- Blog de estética de páginas Web, accedido en 2022
  - <https://www.neolo.com/blog/diseno-web-inspiracion.php>

- Adobe, «Adobe Photoshop,», accedido en 2022
  - <https://www.adobe.com/es/products/photoshop.html>
- GitHub, accedido en 2022
  - <https://github.com>