



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Facultat d'Informàtica de Barcelona



# **Diseño e implementación de una web app para la guardería "El segon cau"**

FACULTAD DE INFORMÁTICA DE BARCELONA  
GRADO EN INGENIERÍA INFORMÁTICA  
ESPECIALIDAD EN INGENIERÍA DEL SOFTWARE

*Pablo Iniesta Romero*

Director: Ernest Teniente López

# Abstract

It is estimated that in Spain, in the year 2019, almost 43 million people or 93% of the population made use of the Internet on a regular basis. It is not unreasonable to say that the Internet has become a permanent phenomenon that influences our lives on the daily.

It is not only people who are influenced. Nowadays, owning a website or domain or even having presence in social networks is almost a mandatory requirement for companies and businesses, because the lack of them can be a great disadvantage compared to their competitors.

The nursery school El Segon Cau is aware of this and that is why it has decided that it is a good moment to modernize itself, counting with a new web application that allows the center to advertise in a simple way, that facilitates the contact with potential clients and that serves the nursery school as a platform for the communication between families and the school, showing the events and happenings that develop along the course in which the students participate.

In this project, this new web app will be developed using the Django framework. The web will have a home page, contact page, several pages to inform about the school and a blog, taking into account at all times the needs and requests of the school.

# Resumen

Se estima que en España, en el año 2019, casi 43 millones de personas o el 93% de la población hacían uso de internet de manera regular. No es descabellado afirmar que internet se ha vuelto un fenómeno permanente que influencia nuestras vidas a diario.

No solo las personas se ven influenciadas. Hoy en día poseer una página web o un dominio o incluso tener presencia en las redes sociales es casi un requisito obligatorio para las empresas y comercios, pues la falta de ellos puede suponer una gran desventaja frente a sus competidores.

La guardería El Segon Cau es consciente de ello y por eso ha decidido que es un buen momento para modernizarse, contando con una nueva aplicación web que permita al centro publicitarse de manera sencilla, que facilite el contacto con clientes potenciales y que sirva a la guardería como plataforma para la comunicación familias-escuela, mostrando los eventos y acontecimientos que se vayan desarrollando a lo largo del curso con el alumnado.

En este proyecto se desarrollará dicha aplicación mediante el framework Django que contará con una página principal, página de contacto, diversas páginas para informar sobre el centro y un blog, teniendo en cuenta en todo momento las necesidades y peticiones del centro escolar.

# Resum

S'estima que a Espanya, l'any 2019, gairebé 43 milions de persones o el 93% de la població feien ús d'internet de manera regular. No és absurd afirmar que internet s'ha tornat un fenomen permanent que influencia les nostres vides diàriament.

No sols les persones es veuen influenciades. Avui dia posseir una pàgina web o un domini o fins i tot tenir presència en les xarxes socials és gairebé un requisit obligatori per a les empreses i comerços, perquè la falta d'ells pot suposar un gran desavantatge enfront dels seus competidors.

La guarderia El Segon Cau és conscient i per això ha decidit que és un bon moment per a modernitzar-se, comptant amb una nova aplicació web que permeti al centre publicitar-se de manera senzilla, que faciliti el contacte amb clients potencials i que serveixi a la llar d'infants com a plataforma per a la comunicació famílies-escola, mostrant els esdeveniments que es vagin desenvolupant al llarg del curs amb l'alumnat.

En aquest projecte es desenvoluparà aquesta aplicació mitjançant el framework Django que comptarà amb una pàgina principal, pàgina de contacte, diverses pàgines per a informar sobre el centre i un blog, tenint en compte en tot moment les necessitats i peticions del centre escolar.

## Índice

1. Contexto .....	9
1.1 Introducción.....	9
1.2 Motivación del proyecto.....	9
1.3 Situación actual .....	10
2. Alcance .....	13
2.1 Objetivos generales.....	13
2.2 Requisitos generales .....	14
2.3 Obstáculos y riesgos .....	16
2.4 Metodología.....	17
2.4.1 Gestión de tareas.....	18
2.4.2 Control de versiones .....	19
3. Estado del arte .....	20
3.1 Sistemas similares .....	20
3.1.1 El Petit Pesolet.....	20
3.1.2 Petits l’Heura .....	21
3.1.3 Llar d’Infants Gaudí .....	21
3.1.4 Conclusiones.....	22
4. Especificación.....	23
4.1 Agentes implicados .....	23
4.2 Casos de uso .....	25
4.2.1 Diagrama de casos de uso.....	25
4.2.2 Especificación de casos de uso.....	26
4.3 Historias de usuario .....	36
4.3.1 Historias de familiares de alumnos.....	37
4.3.2 Historias de clientes potenciales.....	42
4.3.3 Historias de empleados del centro .....	46
4.4 Requisitos no funcionales .....	49
4.4 Modelo conceptual de datos .....	51
4.4.1 Restricciones textuales de integridad .....	54
5. Entorno de desarrollo.....	55
5.1 Sistemas de desarrollo web .....	55
5.1.1 Sistemas de gestión de contenidos .....	55
5.1.2 Conclusiones.....	56

5.1.3 Web Frameworks.....	56
5.1.4 Conclusiones.....	57
5.2 Tecnología y herramientas utilizadas .....	58
6. Diseño.....	60
6.1 Partes del sistema .....	60
6.2 Arquitectura .....	60
6.3 Patrón Model-Template-View .....	61
6.4 Diseño de una funcionalidad con M-T-V .....	63
6.5 Diseño de los models .....	64
7. Implementación .....	65
7.1 Estructura del proyecto.....	65
7.2 Models .....	68
7.2.1 Usuarios .....	68
7.2.2 Aviso.....	69
7.2.3 Blog Post .....	70
7.2.4 Comentario .....	72
7.3 Views & URLs .....	73
7.3 Templates .....	76
8. Validación.....	85
8.1 Modo Debug.....	85
8.2 Validación de funcionalidades .....	85
9. Planificación temporal.....	87
9.1 Planificación temporal inicial.....	87
9.1.1 Fases del proyecto y definición de tareas .....	87
9.1.2 Gantt inicial .....	91
9.1.2 Gestión del riesgo .....	92
9.2 Cambios respecto a la planificación inicial.....	92
9.2.1 Gantt final.....	94
10. Estimación de costes.....	95
10.1 Recursos humanos .....	95
10.2 Recursos materiales .....	96
10.3 Recursos generales .....	96
10.4 Contingencias .....	97
10.5 Coste total.....	97

11. Sostenibilidad .....	98
11.1 Dimensión económica .....	98
11.2 Dimensión medioambiental .....	99
11.3 Dimensión social .....	99
12. Regulaciones y legislación .....	100
13. Conclusiones.....	101
13.1 Futuro .....	102
13.2 Competencias técnicas.....	102
14. Referencias .....	104

## Índice de figuras

Ilustración 1: Ejemplo de mal diseño gráfico .....	11
Ilustración 2: Ejemplo de mala práctica en el blog. ....	12
Ilustración 3: Esquema del funcionamiento de Scrum. ....	17
Ilustración 4: Ejemplo de gestión de un sprint en Taiga. ....	18
Ilustración 5: Diagrama del uso de ramas con Git.....	19
Ilustración 6: Esquema del uso de Git con GitHub. ....	19
Ilustración 7: Página de inicio de El Petit Pesolet. ....	20
Ilustración 8: Página de inicio de Petits l’Heura. ....	21
Ilustración 9: Sección del Blog de la página de la llar d’infants Gaudí. ....	21
Ilustración 10: Diagrama de cebolla de los stakeholders. ....	24
Ilustración 11: Casos de uso de los usuarios estándar .....	25
Ilustración 12: Casos de uso de los usuarios administradores.....	25
Ilustración 13: Estructura de la clasificación de historias de usuario.....	36
Ilustración 14: Diagrama UML del modelo conceptual de datos .....	53
Ilustración 15: Logotipo de WordPress .....	55
Ilustración 16: Logotipo de Blogger.....	56
Ilustración 17: Logotipo de Ruby on Rails.....	57
Ilustración 18: Logotipo de Django.....	57
Ilustración 19: Partes del sistema .....	60
Ilustración 20: Estructura Cliente-Servidor.....	62
Ilustración 21: Diagrama del patrón MTV en Django.....	62
Ilustración 22: Diagrama de secuencia ver blog clase Bambis. ....	63
Ilustración 23: Esquema de los models. ....	64
Ilustración 24: Estructura de un Django Project.....	65
Ilustración 25: Estructura resumida del proyecto .....	66
Ilustración 26: Implementación del model Aviso.....	69
Ilustración 27: Campo de tipo RichTextField. ....	70
Ilustración 28: Implementación del model Blog Post. ....	70
Ilustración 29: Implementación del model Comentario. ....	72
Ilustración 30: Como se relacionan los diferentes elementos de Django.....	73
Ilustración 31: Asociaciones URL-View.....	74
Ilustración 32: Jerarquía de templates .....	76
Ilustración 33: Mock up de la plantilla base.....	77
Ilustración 34: Mock up template Home, sección de información.....	78
Ilustración 35: Mock up template Home, sección de avisos. ....	79
Ilustración 36: Mock up template instalacions. ....	80
Ilustración 37: Mock up template cuina. ....	81
Ilustración 38: Mock up template Blog. ....	82
Ilustración 39: Mock up template Blog Post .....	83

Ilustración 40: Mock up template contacte .....	84
Ilustración 41: Pantalla de error del modo debug.....	85
Ilustración 42: Ejemplo de una issue en Taiga.....	86
Ilustración 43: Diagrama Gantt inicial .....	91
Ilustración 44: Diagrama Gantt final .....	94

## Índice de tablas

Tabla 1: Tabla de tareas del proyecto con su duración y dependencias .....	90
Tabla 2: Tabla de tareas del proyecto de la planificación final .....	93
Tabla 3: Horas y sueldos netos por rol y tarea a realizar.....	96
Tabla 4: Contingencias del proyecto .....	97
Tabla 5: Coste total del proyecto.....	97



# 1. Contexto

## 1.1 Introducción

El proyecto *Diseño e implementación de una web app para la guardería "El segon cau"* se presenta como Trabajo Final de Grado (TFG) con modalidad A en Ingeniería Informática por la Facultad de Informática de Barcelona, Universidad Politécnica de Cataluña. Este trabajo en concreto pertenece a la mención en Ingeniería del Software y se centrará en el ámbito del desarrollo de aplicaciones y servicios web.

Este proyecto pretende desarrollar una aplicación web para un centro escolar, buscando cubrir dos grandes necesidades para la escuela: darse a conocer ofreciendo información sobre el mismo y servir como herramienta para la comunicación entre las familias y el centro.

## 1.2 Motivación del proyecto

Durante la introducción e incluso en el título se hace mención a una empresa, El Segon Cau. Es importante, antes de empezar a hacer hincapié en los diversos aspectos que conforman este trabajo, contextualizarla pues es la base sobre la que gira todo el proyecto y sin ella, éste no existiría.

El Segon Cau es una pequeña guardería privada situada en el barrio de Gràcia, en Barcelona, que cuenta con una gran experiencia en el campo de la educación infantil con más de 50 años de historia a sus espaldas. Una de las características que define al centro es su trato familiar y cercano con sus clientes, ofreciendo atención personalizada para así adaptarse en la mayor medida posible a las necesidades de las diferentes familias y alumnos. En la actualidad cuenta con 5 empleados y acoge a más de 40 infantes entre 0 y los 3 años de edad.

El proyecto nace de la necesidad que El Segon Cau tiene para actualizarse y modernizarse, pues no hacerlo puede suponer una gran desventaja frente a la competencia. En los últimos 10 años, han ido aparecido un gran número de guarderías tanto privadas como públicas en el barrio de Gràcia y alrededores, aumentando de manera significativa la competitividad en el sector. Estos nuevos centros son más propensos a utilizar nuevas tecnologías, aplicándolas tanto para la docencia como para el márketing y el ámbito empresarial, aventajándolos de aquellos con un carácter más tradicional.

El Segon Cau, al ser consciente de ello, decide que una de las mejores formas para afrontar éste fenómeno es mediante la creación de una aplicación web totalmente nueva que permita al centro ofrecer información detallada sobre el mismo de manera clara y

accesible, como por ejemplo los datos de contacto, su localización, sus instalaciones... Así como ofrecer a los padres un lugar que les permita ver las diferentes actividades en las que participan sus hijos e hijas durante el día a día. Además, la aplicación web actuará como primer punto de contacto entre el centro y las familias interesadas en él.

Además de las posibles implicaciones monetarias, tras la implementación de la nueva aplicación web encontramos un aspecto que, a veces, puede ser pasado por alto aun pudiendo llegar a ser incluso más importante que el anterior, pues su valor es verdaderamente incalculable. Este aspecto es el factor humano. El concepto de una guardería está fuertemente ligado con un sentimiento que, por parte de los padres, puede llegar a ser difícil de afrontar y superar: dejar a sus pequeños, posiblemente el aspecto más importante de sus vidas, en manos de personas ajenas o desconocidas. Por ello, es muy importante dotar a los padres y madres de un sentimiento de tranquilidad y seguridad y, en definitiva, hacerles saber que no tienen nada de qué preocuparse al dejar a sus hijos en manos de El Segon Cau. Los profesores y empleados del centro son una parte crucial a la hora de transmitir este sentimiento. Su objetivo principal es brindar toda la atención que sea necesaria a los niños para atender a todas sus necesidades y así garantizar su cuidado y desarrollo. Por tanto, todo su tiempo y esfuerzo debería dedicarse a los alumnos y a los aspectos relacionados con ellos y no ser invertido en “pelearse” con una página web. Con la nueva aplicación web se pretenderá facilitarles la vida a los profesores, haciendo que sea muy sencilla de utilizar y actualizar para que se puedan centrar en lo que al fin y al cabo más importa, los niños.

La aplicación web, como la guardería, deberá poder actualizarse con el tiempo. Teniendo esto en cuenta, a la hora de diseñar e implementar el sistema se procurará que la información sea lo más sencilla de modificar o añadir posible, haciendo que cada funcionalidad sea independiente de las demás. También es importante dotar a la empresa y a sus empleados de las herramientas necesarias para monitorizar, modificar y añadir información sin la necesidad de depender del desarrollador. Este aspecto es vital a la hora de informar sobre las actividades que vayan ocurriendo a lo largo del tiempo en la guardería, pues serán las propias profesoras las encargadas de hacerlo.

Como exalumno de la guardería, siempre he tenido una relación muy cercana con las diferentes empleadas del centro. Desde que decidí cursar la carrera de ingeniería informática me preguntaban, medio bromeando, si sería yo quien les haría la página web. El hecho de tener que hacer un TFG me brinda la oportunidad perfecta para llevar a cabo esta tarea, pudiendo demostrar y aplicar los conocimientos adquiridos a lo largo del grado, con especial interés en el apartado de ingeniería de software y diseño e implementación de tecnologías y aplicaciones web.

## 1.3 Situación actual

Este proyecto no supone el primer intento que El Segon Cau realiza con tal de iniciarse en el basto mundo de internet. En 2016 y por iniciativa de las propias empleadas, se creó un blog [\[1\]](#) para la guardería mediante la herramienta de publicación de blogs gratuita Blogger [\[2\]](#). Toda aplicación web de calidad debería cumplir con unos ciertos

principios o bases que permitan a la aplicación llevar a cabo su función lo mejor posible, en nuestro caso será informar. Principios como:

- Ofrecer contenido de calidad: Nuestro contenido debe ser actual, veraz, relevante, y de fácil acceso para nuestros usuarios.
- Interfaz amigable: Cuanto más sencilla de utilizar sea la página, mejor. Todo tipo de persona deberá ser capaz de identificar con facilidad los diferentes contenidos que se vayan mostrando en pantalla.
- Navegación fácil e intuitiva: Los usuarios deben poder navegar por la web sin problema ni conocimiento previo de ningún tipo.
- Buen diseño gráfico: La página debe ser agradable a la vista, limpia y funcional. También debería adaptarse a la temática escolar.
- Optimización: Es importante que la página sea rápida y ágil a la hora de cargar contenido. Cuanto más tarde una página más probable es que el usuario la abandone.

Por desgracia, el blog actual no cumple muy bien con ellos. Al ser creado por las propias profesoras del centro, cuya experiencia previa con herramientas de creación de sitios web es casi nula, las carencias son evidentes. El principal problema que presenta la página web es el difícil acceso a toda la información que el profesorado quiere dar a conocer sobre su centro ya sea por la mala distribución de los contenidos, por la falta de visibilidad y coherencia en la estructura del blog o por un diseño gráfico no agradable a la vista que dificulta la retención de la atención del usuario. Otro aspecto a tener en cuenta es la prácticamente nula interacción entre los padres o clientes en potencia que visitan la página y el centro.

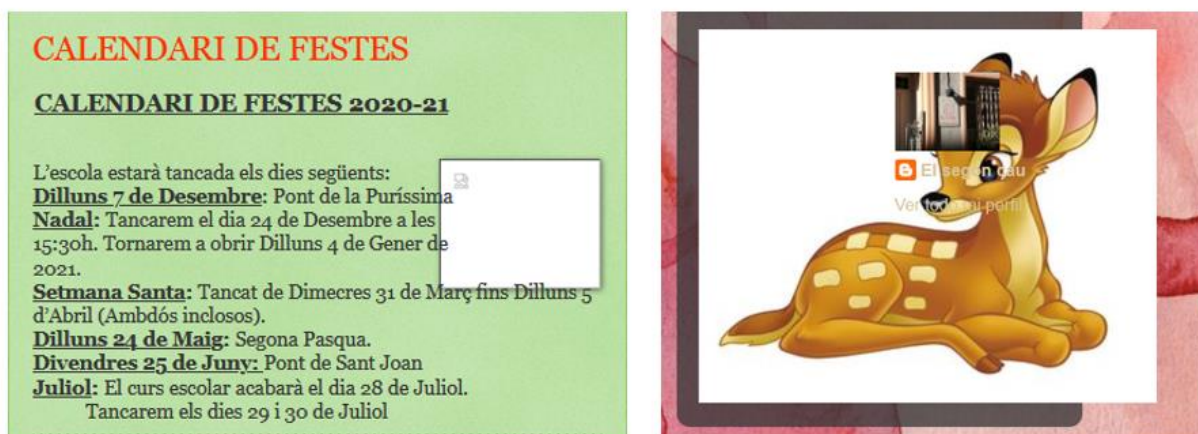


Ilustración 1: Ejemplo de mal diseño gráfico. Las imágenes y elementos decorativos no se adaptan correctamente a la estructura de la página.

**PLA DE TREBALL DE LA LLAR EN CONFINAMENT:**  
En cas de confinament parcial o tancament de l'escola, es procedirà a mantenir actiu el contacte amb les famílies via whatsapp setmanal, i la creació de contingut pedagògic amb propostes per a les famílies i els més petits i es continuarà en la mesura que es pugui amb la programació que es tenia prevista pel curs.  
Previsió d'alumats que assistirà al centre:  
29 alumnes.  
Les famílies es faran responsables de l'estat de salut dels seus fills/es comproment-se a no portar-lo a la llar d'infants en cas que presenti simptomatologia compatible amb la COVID-19 o l'hagi presentat en els darrers dies.  
Nombre d'alumnes:  
L'alumnat del centre:  
Nombre de grups:  
14 dies i a comunicar-ho immediatament als responsables del centre educatiu per tal de poder prendre les mesures oportunes.  
De forma excepcional es podrà reincorporar alumnat amb malaltia crònica d'elevada complexitat amb declaració responsable dels progenitors o legals representants que acredita que han valorat amb el seu metge/metgessa o pediatre/a la idoneïtat de reprendre l'activitat escolar.  
**ORGANITZACIÓ DE GRUPS D' ALUMNES, PROFESSIONALS I ESPAIS:**  
Grups Alumnes:P-0/P-1/P-2  
Docents /Espai  
Estable/Temporal  
Acollida matí/Acollida tarda:  
Ullan Olga, Reina Olga, Alvarez Anna, Romero Montserrat i Giraldez Patricia.  
Aula, Po Aula P1, Aula P2, Aula multiusos.  
Aquests grups seran estables i fixes per poder tenir la traçabilitat de contactes en cas de contagi. L

Il·lustració 2: Ejemplo de mala pràctica en el blog. Una gran quantitat de text sense estructura dificulta la lectura i no atrau al usuari.

Con este trabajo se pretenderá poner solución a todos los problemas anteriormente descritos mediante la creación de una aplicación web totalmente nueva que se adapte en la mayor medida posible a las necesidades del centro.

## 2. Alcance

Con tal de garantizar el buen desarrollo y éxito del proyecto y teniendo en cuenta que el periodo de tiempo para diseñarlo e implementarlo está prefijado y es reducido, es necesario definir el alcance de este identificando los objetivos principales y sus requisitos.

### 2.1 Objetivos generales

Detallando un poco más a fondo el proyecto nos podemos hacer una idea de los objetivos que pretenderá cumplir.

La aplicación web que se desarrollará deberá ser accesible de manera gratuita para toda aquella persona que desee visitarla, sin importar su localización, la fecha o la hora del día. Como uno de los objetivos principales es informar al mayor número de personas posibles, su diseño y uso deberá ser lo más simple e intuitivo posible. Toda persona debería poder utilizarla al 100% sin ningún tipo de entrenamiento o conocimiento previo, incluso personas con diferentes niveles de discapacidad.

La aplicación contará con un “home” o página de inicio. Esta será la primera página a la que un usuario accederá una vez haya accedido a nuestra aplicación. Al ser el home el primer contacto usuario-centro, toda la información relevante sobre la guardería deberá ser mostrada en él. Información como que es El Segon Cau, su metodología y filosofía a la hora de enseñar, horarios y todo tipo de información que se considere relevante.

Otro aspecto importante en el que muchas familias se fijan a la hora de buscar una guardería para sus hijos son las instalaciones. La web contará con un apartado propio para las instalaciones, donde se nombrará y describirá de manera breve cada una de ellas y se ilustrarán mediante contenido multimedia.

Otro punto de gran importancia a la hora de tratar con infantes es la alimentación. No todos los alumnos tienen la misma edad y las mismas necesidades y es por ello por lo que el centro ofrece diversos menús según la edad de los alumnos y posibles adaptaciones a alergias, dietas... Estos menús y opciones estarán reflejados en otra sección de la aplicación web.

También será necesaria una página de contacto. En este apartado se ofrecerá toda la información necesaria para contactar con la empresa como el teléfono y la dirección de correo electrónico. Además se mostrará la ubicación del centro mediante alguna herramienta de geo-localización como Google Maps. Con tal de facilitar el contacto usuario-centro, se dará la opción de rellenar un formulario donde se podrá declarar un correo de contacto, el motivo de la consulta así como una explicación más extensa de la misma. Esta consulta se enviará automáticamente al correo de la guardería desde el que se le responderá al usuario.

Un objetivo que se ha mencionado previamente es el informar sobre el día a día de cada una de las clases de la guardería. La mejor manera para hacerlo es mediante un

blog. Cada clase tendrá su sección propia, pues las actividades que realizan los alumnos dependen de su edad, por tanto se habrá 3 secciones diferentes. Cada blog estará compuesto por entradas o “posts”. Cada post tendrá un título y una portada, texto opcional para explicar la actividad si es necesario y hasta 10 fotos para ilustrarla. Como todo blog que se precie, también se contará con la posibilidad de dejar comentarios en cada entrada. Estos comentarios serán moderados por el centro y solo se mostraran una vez hayan sido aceptados. Con esta funcionalidad se espera aumentar la interacción entre los padres y la guardería.

El hecho de contar con un blog y poder dejar comentarios implica la necesidad de tener usuarios. Registrarse como usuario en la aplicación web será gratuito y toda persona que quiera podrá hacerlo. Se contará con diferentes tipos de usuario con características diferentes. El usuario básico no contará con ningún tipo de permiso especial, podrá hacer uso de la aplicación de manera normal y podrá crear y eliminar sus propios comentarios. Otro tipo de usuario se corresponde con los empleados del centro. El tipo de usuario empleado tendrá acceso a algunas de las funcionalidades del panel de administración, donde se podrán crear posts y gestionar los posts y verificar comentarios. Al dotar a los empleados de esta herramienta, se garantiza el objetivo de ir actualizando el contenido sin depender del desarrollador. Por último está el usuario administrador, al que solo tendrá acceso el desarrollador. Este usuario tiene acceso a todas las funciones del panel de administración como la gestión de usuarios e información sobre la aplicación.

Será posible acceder, desde cada una de estas funcionalidades, a todas las demás en todo momento gracias a un menú o barra de navegación presente en cada una de las páginas haciendo la navegación una tarea sencilla.

## 2.2 Requisitos generales

Tras haber descrito con más detalle la aplicación y una vez habiendo identificado los objetivos que se pretenden cumplir, podemos dividirlos en requisitos que nos ayudaran a entender de una manera más clara que funcionalidades cubrirá cada uno de ellos.

### **Permitir al centro ofrecer información sobre él y darse a conocer**

- Mostrar información de contacto como el correo electrónico y teléfono.
- Ofrecer una breve introducción sobre la escuela.
- Mostrar la ubicación del centro mediante alguna herramienta de geo-localización como Google Maps.
- Especificar las instalaciones del centro y mostrarlas con imágenes.
- Especificar los horarios de la Guardería.
- Especificar posibles eventos especiales o noticias importantes.
- Informar sobre la comida que se sirve en el centro.

### **Permitir la identificación de usuarios, ya sean empleados del centro, familiares de los alumnos o familias interesadas en la guardería.**

- Contar con un sistema de usuarios con sus funciones de registro, log-in y log-out.
- Dotar a cada tipo de usuario de unas características u otras dependiendo de su función. Un usuario profesor podrá hacer publicaciones y moderar comentarios, un usuario administrador podrá acceder a todo el contenido.

### **Permitir a las profesoras comunicar el día a día de las clases y de los alumnos a las familias.**

- Contar con una sección distinta para cada clase.
- Permitir la creación, edición y eliminación de entradas o “posts” en cada una de esas secciones.
- Permitir descripciones de texto, fotos y videos en cada entrada para cada clase.
- Permitir la creación y eliminación de comentarios en los posts.
- Permitir la moderación de comentarios. Un comentario no se mostrará al público si no ha sido aceptado previamente por una cuenta administradora.

### **Facilitar la comunicación y el contacto con el centro a través de la aplicación web.**

- Permitir contactar con el centro mediante un formulario.
- Permitir el contacto con el centro tanto si se está registrado como si no se está, permitiendo declarar un correo electrónico.
- Permitir declarar un asunto o el motivo del contacto y una descripción extensa del mismo.

A parte de los requisitos funcionales, también debemos tener en cuenta los requisitos no funcionales que afectarán a la aplicación.

- Requisito de apariencia
- Requisito de facilidad de uso
- Requisito de velocidad y latencia
- Requisito de fiabilidad y disponibilidad
- Requisito de producción
- Requisito de privacidad y seguridad
- Requisito de legalidad

Veremos todos estos requisitos con más detalle más adelante en el punto 4 que trata la especificación del sistema.

## 2.3 Obstáculos y riesgos

Todo proyecto de software conlleva sus riesgos. Es importante identificar posibles riesgos u obstáculos antes de empezar a implementar el proyecto y estimar su impacto. De esta manera sortearlos con éxito será más rápido y sencillo en caso de que acaben ocurriendo.

**Comunicación con El Segon Cau:** Es cierto que el hecho de que este trabajo sea un TFG de modalidad A, puede suponer una mayor libertad a la hora de diseñar y especificar el proyecto comparado con uno de modalidad B, donde empresas realizan el proyecto siguiendo encarecidamente los requisitos del cliente. No obstante, esto no quiere decir que la comunicación con el centro no sea importante. Al fin y al cabo la aplicación web deberá adaptarse al máximo a las diversas peticiones e ideas de la guardería, por tanto una buena comunicación entre el desarrollador y el centro es vital para tener siempre la misma visión de la aplicación en las diferentes partes del proyecto.

**Cambios en el proyecto:** Otro riesgo a tener en cuenta y que también está relacionado con la comunicación con el centro son los posibles cambios en la especificación que puedan ocurrir en el transcurso del proyecto. El centro tendrá la opción durante todo momento de ofrecer su opinión, pudiendo llegar a modificarse, añadirse o eliminarse funcionalidades durante el proceso de desarrollo.

**Malas estimaciones i/o planificación:** La existencia de una fecha límite de entrega para el trabajo puede llegar a constituir un riesgo. Un mal análisis del alcance del trabajo o una mala distribución del trabajo a la hora de implementar la solución podría hacer peligrar el proyecto, pues no estaría listo a tiempo.

**Problemas técnicos:** El desarrollador solo cuenta con un equipo informático propio para trabajar. Cualquier tipo de mal funcionamiento que pudiese llegar a hacer que el ordenador dejase de funcionar podría llegar a afectar a la planificación temporal del proyecto.

**Inexperiencia:** A lo largo del proyecto se hará uso de diversas tecnologías y frameworks con las que el desarrollador no tiene mucha experiencia o incluso experiencia nula. El hecho de necesitar aprender a utilizar todo desde cero puede llegar a hacer que el tiempo de desarrollo sea mayor de lo estimado.

**COVID-19:** Por desgracia el coronavirus también puede suponer un riesgo para el proyecto, especialmente al tratarse de un centro escolar. Como guardería, El Segon Cau debe poner especial atención en la prevención y lucha contra el Covid-19. La aparición de casos positivos supondría dificultades para la guardería, pudiendo hacer que el proyecto dejase de ser un aspecto de importancia inmediata para el centro, perjudicando el buen desarrollo de la aplicación web. Otro punto a tener en cuenta es la limitación en el contacto directo entre personas. Esto puede dificultar la comunicación entre el desarrollador y la escuela, aunque con las varias herramientas para la comunicación en línea disponibles no debería llegar a suponer ningún problema importante.



## 2.4 Metodología

El hecho de que el proyecto de creación de una aplicación web para El Segon Cau sea a su vez un trabajo de final de grado implica que el desarrollo del trabajo debe estar lo más pautado posible y que debe realizarse en un tiempo prefijado, 4 meses aproximadamente o la duración de un cuatrimestre. Por ello se ha decidido optar por seguir una metodología ágil, común en los proyectos de desarrollo de software.

Tras analizar las principales metodologías ágiles, se llegó a la conclusión que el método Scrum [3] con ciertas adaptaciones ayudaría a cumplir los objetivos del proyecto a la perfección y se amoldaría muy bien a la filosofía de desarrollo del proyecto.

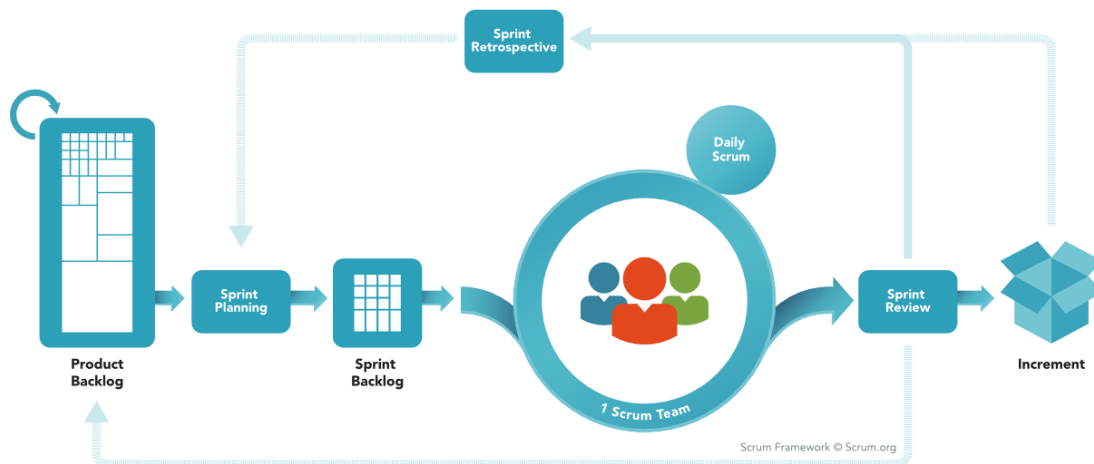


Ilustración 3: Esquema del funcionamiento de Scrum.

Gracias a la metodología Scrum el proyecto se irá desarrollando e implementando de manera continua e iterativa a lo largo del cuatrimestre. Se dividirá el periodo de tiempo para el desarrollo en partes más pequeñas, llamadas iteraciones o sprints, en las cuales se desarrollará un aspecto específico de la aplicación. Al inicio de cada sprint se realizará el sprint planning, donde se decidirá que funcionalidades se desarrollarán. Una vez acabado el sprint planning obtendremos el sprint backlog, una lista de ítems y tareas que se deberán realizar a lo largo del sprint. Al final de cada iteración, mediante un sprint review, se evaluará el progreso y, en caso de ser necesario, se concertará una reunión con el director del TFG para discutir la dirección del proyecto con tal de poder aplicar cambios en la planificación o la metodología para adaptar el proyecto a los posibles contratiempos que puedan surgir.

Scrum está enfocado principalmente a equipos pequeños de desarrollo de software, pero al estar el equipo de desarrollo de este proyecto formado por una única persona es necesario hacer unas pequeñas adaptaciones. Según Scrum, el equipo de desarrolladores debería realizar una reunión breve a diario para coordinar el trabajo. Al ser una única persona trabajando esto no es necesario. Además en Scrum existe la figura del Scrum Master, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. Normalmente el Scrum Master es independiente al equipo de desarrolladores pero en este caso la misma persona hará ambas funciones.

Para el desarrollo del proyecto se han planteado cuatro sprints diferentes de unos 20 días de duración aproximadamente. En el primer sprint se preparará el entorno de trabajo y una primera estructura de la web para poder empezar a trabajar. También se implementará el sistema de usuarios pues son necesarios para implementar las siguientes funcionalidades. En el sprint 2 se implementará el sistema de blog de la guardería. El tercer sprint se centrará en el aspecto informativo de la aplicación web y el último estará enfocado a mejorar la interfaz gráfica.

## 2.4.1 Gestión de tareas

Existen un gran número de herramientas para facilitar la gestión de tareas y el seguimiento del trabajo a realizar en cada sprint. En este proyecto se ha decidido utilizar Taiga [4] por ser una herramienta potente, gratuita, accesible online y porque ya se ha utilizado varias veces a la hora de gestionar trabajos durante la carrera.

Taiga es una herramienta de código libre diseñada para la planificación y desarrollo de proyectos de software que sigan metodologías ágiles, en nuestro caso Scrum. Al crear un proyecto en Taiga podemos dividirlo en sprints. Para cada sprint se asignarán las tareas que será necesario completar. Estas tareas pueden ser puntuadas con un valor dependiendo de la dificultad o número de horas estimadas que llevará acabo realizarla. Cada tarea a realizar puede ser dividida en sub-objetivos, haciendo la organización mucho más dinámica y granular. Cada uno de estos sub-objetivos se puede clasificar según su estado de completitud: Nuevo, En Progreso, Listo Para Testear y Cerrado.

SPRINT 1 WEBAPP EL SEGO... 21 OCT 2020-10 NOV 2020

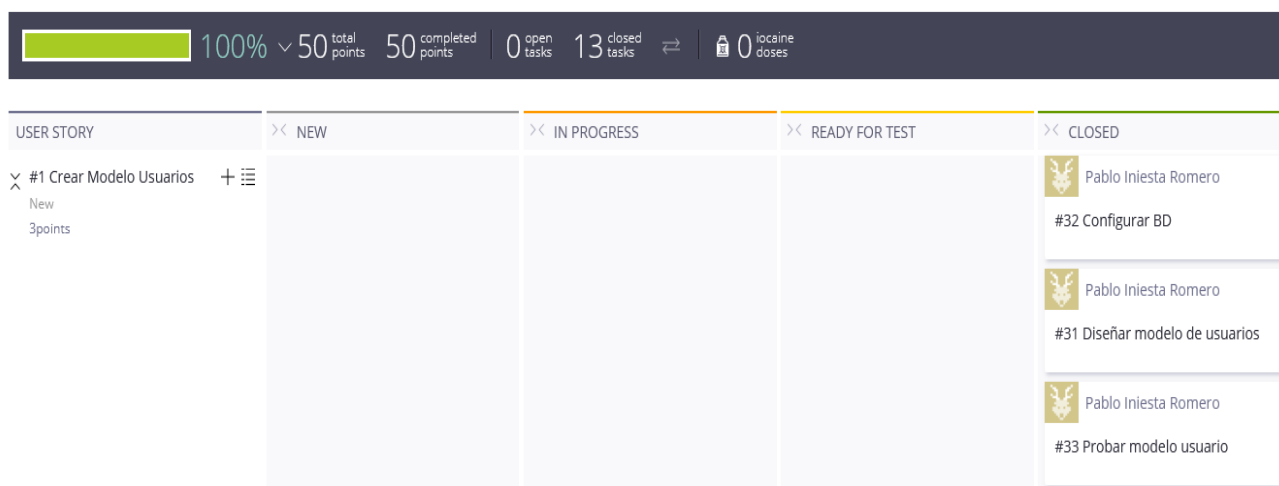


Ilustración 4: Ejemplo de gestión de un sprint en Taiga.

Dependiendo de la cantidad de tareas terminadas o “cerradas” en un sprint podremos observar la evolución del proyecto. Taiga también ofrece la opción de asignar tareas a miembros del equipo de desarrollo específicos pero en nuestro caso no será de mucha utilidad pues todas serán realizadas por la misma persona.

## 2.4.2 Control de versiones

Todo proyecto de software necesita una herramienta para el control y la actualización de código según el proyecto avance. Para este proyecto se utilizarán 2 de las herramientas más usadas en el control de versiones, Git y GitHub.

Git [5] es una herramienta de control de versiones que permite realizar todo tipo de operaciones para recuperar o enviar datos, que en nuestro caso es el código del proyecto. Git permite la coordinación entre desarrolladores mediante un sistema de “branches” o ramas. La rama principal o “master” será la que contenga el código plenamente operativo y definitivo. Cada desarrollador creará una sub-rama local con los contenidos de la rama principal donde podrá realizar los cambios necesarios o añadir nuevas funcionalidades. Una vez estos cambios o funcionalidades hayan sido terminados y testeados se añadirán a la rama principal, pasando a ser definitivos.

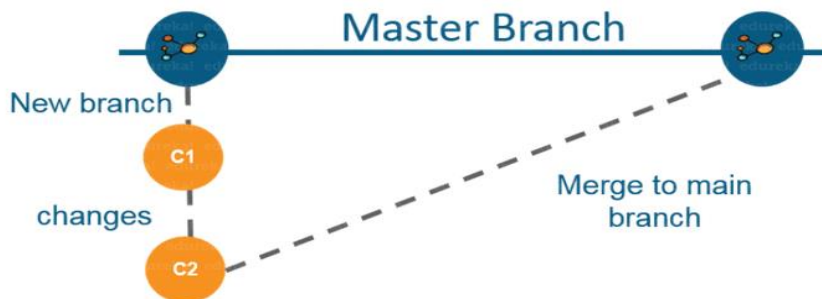


Ilustración 5: Diagrama del uso de ramas con Git.

Por otro lado tenemos GitHub [6], un portal para alojar proyectos que utiliza el sistema de control de versiones Git. Todos los ficheros que hacen posible la ejecución de la aplicación estarán alojados en un repositorio de GitHub. Gracias a este repositorio online se podrá acceder al proyecto en cualquier dispositivo, por tanto no será necesario depender de ningún ordenador para desarrollarlo.

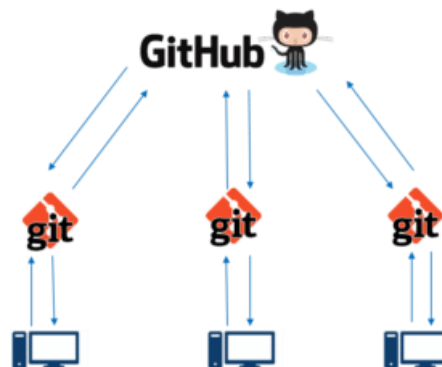


Ilustración 6: Esquema del uso de Git con GitHub.

## 3. Estado del arte

Antes de empezar a implementar el proyecto es importante analizar las tecnologías existentes y soluciones parecidas a la que se debe desarrollar. Con este análisis se pretende adquirir un mejor entendimiento del sector al que pasará a formar parte la aplicación web una vez haya sido completada. Por una parte se estudiarán otros sistemas similares, en concreto las aplicaciones web de guarderías cercanas al Segon Cau y por otra se estudiarán diversas tecnologías para el desarrollo web.

### 3.1 Sistemas similares

En este punto se analizarán las aplicaciones web de las diferentes guarderías cercanas al Segon Cau y que por tanto suponen una competencia directa. Con este análisis se pretende ver posibles soluciones a los problemas expuestos anteriormente y posibles oportunidades de mejora que permitan destacar sobre el resto.

#### 3.1.1 El Petit Pesolet

El Petit Pesolet es una guardería muy cercana al Segon Cau, separadas por poco más de una manzana de distancia. En su web podemos ver diversos apartados donde se explica la filosofía de la escuela, su localización, sus datos de contacto y un formulario para contactar con el centro, también se ofrecen imágenes del centro y sus instalaciones y los diferentes servicios que se ofertan como casales y fiestas.



Ilustración 7: Página de inicio de El Petit Pesolet.

Se echa a faltar algún tipo de contenido que ilustre o explique el día a día a día de la escuela. La información expuesta en la web es suficiente para generar una idea clara de cómo es el centro pero no ayuda mucho a esclarecer cómo va a ser la experiencia de los niños una vez estén en clase.

### 3.1.2 Petits l'Heura

Petits l'Heura es otra guardería situada cerca del Segon Cau. En su aplicación web, al igual que con la del Petit Pesolet, se nos ofrece una breve descripción del centro, datos de contacto, su localización, un formulario de contacto, imágenes de las instalaciones... Un aspecto único que tiene esta aplicación web es la opción de dar un número de teléfono para que el centro se ponga en contacto contigo de manera gratuita.



Ilustración 8: Página de inicio de Petits l'Heura.

Tal y como pasaba con El Petit Pesolet, falta una sección que ayude a ver las actividades que se llevan a cabo en el centro. Otro punto interesante que podría llegar a jugar en contra del centro es que en ciertas secciones como el horario o los servicios se ofrece una información muy poco detallada y es necesario contactar con el centro mediante un formulario para saber más.

### 3.1.3 Llar d'Infants Gaudí

A parte de ofrecer las secciones comunes entre todas las aplicaciones web de guarderías como información sobre el centro, datos de contacto, etc. La Llar d'Infants Gaudí sí que cuenta con una sección enfocada al desarrollo diario de los niños en forma de un blog. Por desgracia, tanto esta como otras cuantas secciones más no están



Ilustración 9: Sección del Blog de la página de la llar d'infants Gaudí.

disponibles si no estás registrado en la web y para estar registrado necesitas ser el familiar de algún alumno.

Lo más probable que en el blog aparezcan imágenes de los alumnos y por tanto sea necesario tener un “bloqueo” para respetar la privacidad de los niños y sus familias. Esto puede llegar a ser un punto en contra pues podría suponer que una familia interesada se decante por el centro tras ver el blog. En la aplicación de El Segon Cau no se mostrará en ningún momento la cara de ningún alumno, por tanto el bloqueo de contenido no será necesario.

### 3.1.4 Conclusiones

Tras visitar y analizar diversas aplicaciones de guarderías y centros infantiles podemos ver que todas comparten una misma estructura y, en mayor o menor medida, tienen un contenido similar. En todas se proporciona información sobre el centro y se da un poco de contexto sobre la empresa. También se proporcionan todos los datos y la información pertinentes para el contacto, comúnmente mediante el uso de un formulario.

Un punto fuerte que puede llegar a marcar la diferencia respecto a las otras webs es tener una sección que actúe como blog para cada una de las clases y que sea de libre acceso.

## 4. Especificación

Antes de empezar con la implementación de la aplicación, es necesario identificar los requisitos que garantizarán que la aplicación incorpora las funcionalidades requeridas y que acabe siendo un producto de calidad. Para poder hacernos una mejor idea de los requisitos necesarios, primero veremos los agentes implicados en él. En segundo lugar se explicaran los requisitos funcionales del sistema en forma de casos de uso e historias de usuario con su esquema pertinente. También es importante definir y justificar los requisitos no funcionales que afectan al sistema. Para finalizar se mostrará y explicara el modelo conceptual de los datos del sistema.

### 4.1 Agentes implicados

En este apartado se definirán los *stakeholders* o las partes interesadas relacionadas con el proyecto, es decir, toda persona, empresa u organismo que tenga relación, ya sea de manera activa o pasiva, con el desarrollo y uso de la aplicación web. Esta tarea es muy importante, ya que una buena identificación y definición de los stakeholders proporcionará conocimiento y ayudará a definir los requisitos que debe alcanzar el sistema.

#### **El Segon Cau**

La empresa El Segon Cau es el stakeholder principal y la mayor interesada en el éxito del proyecto. Gracias al diseño y la implementación de la nueva aplicación web, El Segon Cau espera aumentar el número de clientes potenciales que se interesan por el centro y así poder generar más beneficios a corto y largo plazo. Además, también esperan que la nueva aplicación web aumente la satisfacción de las familias de los niños ya matriculados y mejore su imagen de cara al exterior así como hacer de la comunicación escuela-familia algo más sencillo. Por todo esto la organización será la mayor beneficiaria del proyecto.

#### **Empleados del centro**

Los propios empleados del centro también forman parte de los stakeholders, pues harán uso de la nueva aplicación web de manera periódica. El proyecto facilitará los aspectos comunicativos entre el centro y las familias y también incrementará la facilidad y la comodidad a la hora de hacerlo respecto a la solución actual. Es necesario remarcar que una parte de la responsabilidad del éxito del proyecto recaerá sobre este grupo ya que serán los propios profesores los que deberán ir actualizando la información respectiva a su clase o actividad de manera periódica.

#### **Familias de los alumnos**

Las familias de los alumnos matriculados en el centro también forman una parte importante de los agentes implicados. Mediante la aplicación web serán capaces de conocer el día a día de la guardería a la que sus hijos acuden y enterarse de posibles

eventos o acontecimientos importantes como fiestas, actividades especiales, horarios, cambios en la guardería...

## Familias que buscan una guardería

Las familias que estén buscando un centro para sus hijos también son beneficiarias directas del proyecto. Con la nueva aplicación web les será más fácil obtener todo tipo de información relacionada con el centro y también les será útil para contactar con el mismo en caso de que quieran obtener una entrevista o información que se ajuste a su situación concreta.

## Alumnos

Los alumnos y alumnos potenciales son también una parte interesada. Es cierto que no harán un uso directo de la aplicación web, son bebés o infantes de como máximo 3 años de edad, pero el hecho de tener una buena aplicación puede marcar la diferencia entre el hecho de acabar atendiendo o no a El Segon Cau, influenciando así su primera etapa en el sistema escolar.

## Desarrollador

Yo, Pablo Iniesta, como único desarrollador, seré una de las partes interesadas en el éxito del proyecto. Seré el encargado de planificar y desarrollar la aplicación web así como de documentar y redactar la memoria del TFG. Si todo sale según lo previsto y el proyecto acaba siendo un éxito, será posible que me gradúe al acabar este cuatrimestre.

## Director del TFG y tutor de GEP

Tanto el director del TFG como el tutor de GEP asignado a este proyecto también son partes interesadas. Supervisarán de manera activa el desarrollo del proyecto y velarán por su correcta ejecución y éxito a lo largo del cuatrimestre.

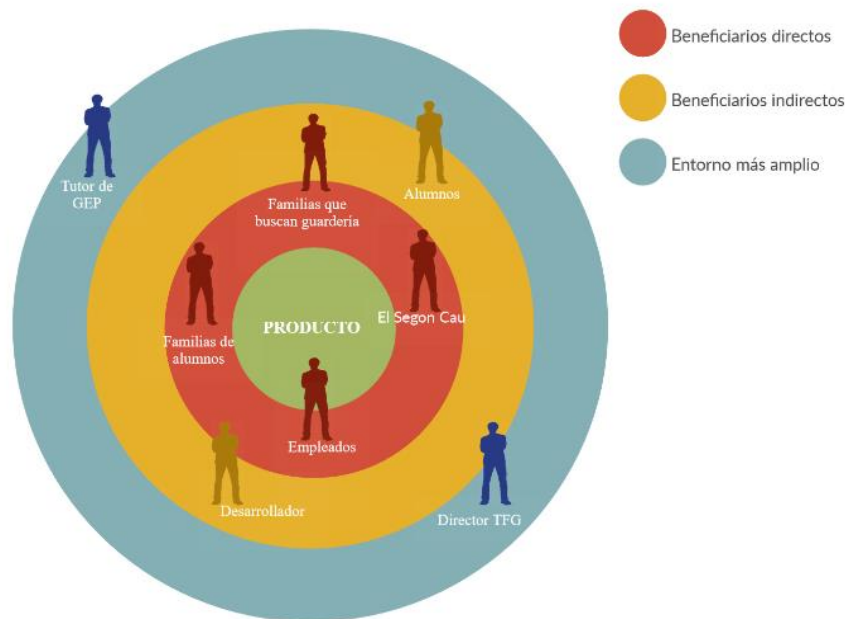


Ilustración 10: Diagrama de cebolla de los stakeholders.



## 4.2 Casos de uso

Una vez vistos los agentes implicados en el proyecto, podemos pasar a especificar los casos de uso que englobarán todas las funcionalidades del sistema.

### 4.2.1 Diagrama de casos de uso

En el diagrama de casos de uso se divide a los usuarios que harán uso del sistema en 3 sub-categorías, dependiendo de su motivación a la hora de acceder a la web. Los familiares de los alumnos, que como su nombre indica, son todos los usuarios que son familia de los niños y niñas matriculados en el centro, estarán más interesados en conocer las actividades y obtener información sobre el día a día de sus hijos e hijas. Los clientes potenciales son todos aquellos padres que estén buscando una guardería para sus hijos. Estarán más interesados en obtener información sobre la guardería como centro educativo. Por último tenemos a los propios empleados del centro cuyo objetivo será añadir el contenido que se necesite, mantener la aplicación actualizada y moderar el contenido según sea necesario.

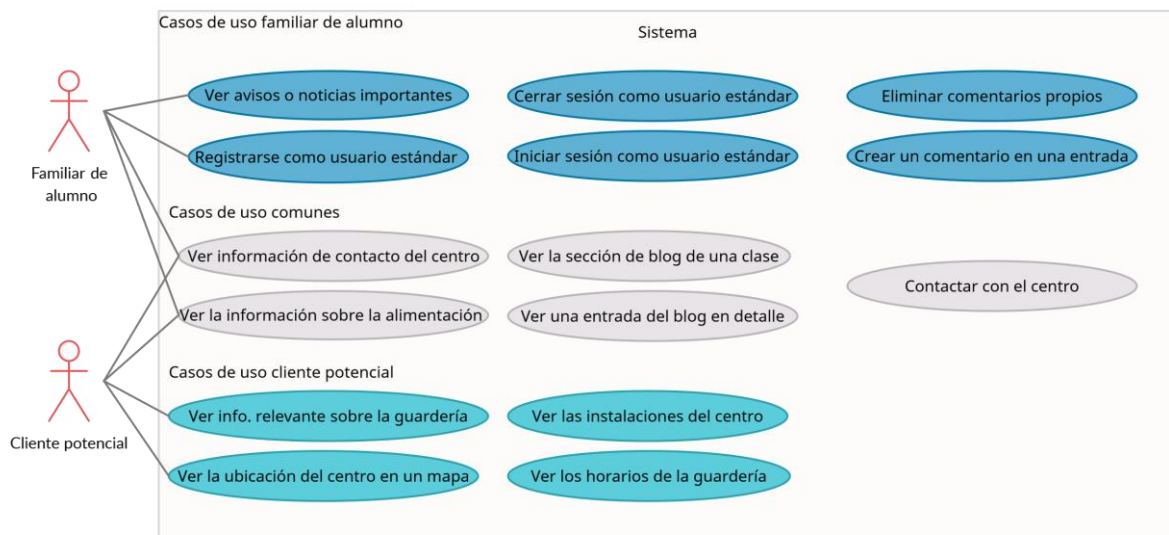


Ilustración 11: Casos de uso de los usuarios estándar



Ilustración 12: Casos de uso de los usuarios administradores

## 4.2.2 Especificación de casos de uso

A continuación se especificarán los casos de uso mostrados en el diagrama del punto anterior. Para cada caso de uso, se especificará su nombre, con que actor o actores principales está relacionado, las precondiciones necesarias para que se lleve a cabo el caso de uso, el escenario principal que garantizará el éxito de la operación y posibles extensiones del escenario principal. Los agruparemos en dos categorías, los casos de uso que hacen referencia a los usuarios estándar, en nuestro caso los clientes potenciales y los familiares de los alumnos, y los usuarios administradores que se corresponden con los empleados del centro.

### **Casos de uso de Clientes potenciales y Familiares de alumnos**

#### *Ver información de contacto del centro*

<b>Caso de uso</b>	Ver información de contacto del centro
<b>Actor principal</b>	Cliente potencial/Familiar de alumno
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
1. Accede a la página web. 2. Selecciona “Contacte” en la barra de navegación. 3. La información de contacto aparece en pantalla.	
<b>Extensiones</b>	
2.a Baja al final de la página. 2.b La información de contacto aparece en la sección final de la página web.	

#### *Ver avisos o noticias importantes*

<b>Caso de uso</b>	Ver avisos o noticias importantes
<b>Actor principal</b>	Familiar de alumno
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
1. Accede a la página web. 2. Baja hasta la sección “Avisos i noticias”. 3. Los avisos i noticias se muestran en pantalla.	
<b>Extensiones</b>	

### *Ver la información sobre la alimentación*

<b>Caso de uso</b>	Ver la información sobre la alimentación
<b>Actor principal</b>	Cliente potencial/Familiar de alumno
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
1. Accede a la página web. 2. Selecciona “Cuina” en la barra de navegación. 3. La información sobre la alimentación aparece en pantalla.	
<b>Extensiones</b>	

### *Ver la información relevante sobre la guardería*

<b>Caso de uso</b>	Ver la información relevante sobre la guardería
<b>Actor principal</b>	Cliente potencial
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
1. Accede a la página web. 2. La información sobre la guardería aparece en pantalla.	
<b>Extensiones</b>	

### *Ver la ubicación del centro en un mapa*

<b>Caso de uso</b>	Ver la ubicación del centro en un mapa
<b>Actor principal</b>	Cliente potencial
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
1. Accede a la página web. 2. Selecciona “Contacte” en la barra de navegación. 3. Un mapa con la localización exacta de la guardería se muestra en pantalla. 4. Presiona en el mapa y es re-direccionado a Google Maps, donde se muestra con más detalle la localización del centro.	
<b>Extensiones</b>	

### *Ver las instalaciones del centro*

<b>Caso de uso</b>	Ver las instalaciones del centro
<b>Actor principal</b>	Cliente potencial
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Accede a la página web.</li><li>2. Selecciona “Instalaciones” en la barra de navegación.</li><li>3. La información sobre las instalaciones aparece en pantalla.</li><li>4. Presiona en las imágenes para verlas con un tamaño mayor.</li></ol>	
<b>Extensiones</b>	

### *Ver los horarios de la guardería*

<b>Caso de uso</b>	Ver los horarios de la guardería
<b>Actor principal</b>	Cliente potencial
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Accede a la página web.</li><li>2. Selecciona “Contacte” en la barra de navegación.</li><li>3. La información sobre los horarios aparece en pantalla.</li></ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"><li>2.a Baja al final de la página.</li><li>2.b La información sobre los horarios aparece en la sección final de la página web.</li></ol>	

### *Registrarse como usuario estándar*

<b>Caso de uso</b>	Registrarse como usuario estándar
<b>Actor principal</b>	Familiar de alumno
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Accede a la página web.</li><li>2. Selecciona “Registrar-se” sobre la barra de navegación.</li><li>3. El formulario de registro aparece en pantalla.</li><li>4. Rellena el formulario de registro con los datos pertinentes.</li><li>5. Es re-direccionado a la página de log-in y se informa que la cuenta ha sido creada correctamente.</li></ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"><li>5.a Los datos introducidos en el formulario no son correctos o no cumplen las restricciones mínimas establecidas.</li><li>5.b Al pulsar “Registrar-se” aparece un mensaje informando sobre los errores.</li></ol>	

### *Iniciar sesión como usuario estándar*

<b>Caso de uso</b>	Iniciar sesión como usuario estándar
<b>Actor principal</b>	Familiar de alumno
<b>Precondición</b>	El usuario posee una cuenta creada previamente
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Accede a la página web.</li><li>2. Selecciona “Login” sobre la barra de navegación.</li><li>3. El formulario de log-in aparece en pantalla.</li><li>4. Rellena el formulario de inicio de sesión con los datos pertinentes.</li><li>5. Es re-direccionado a la página de inicio.</li><li>6. Puede comprobar que ha iniciado sesión correctamente. Donde aparecían los botones de log-in y registro ahora aparece el nombre de usuario.</li></ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"><li>5.a Los datos introducidos en el formulario no son correctos.</li><li>5.b Al pulsar “Fer Login” aparece un mensaje informando sobre los errores.</li></ol>	

### *Cerrar sesión como usuario estándar*

<b>Caso de uso</b>	Cerrar sesión como usuario estándar
<b>Actor principal</b>	Familiar de alumno
<b>Precondición</b>	El usuario posee una cuenta creada previamente y ha iniciado sesión en ella.
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Selecciona “Logout” sobre la barra de navegación.</li><li>2. Se cierra la sesión actual.</li><li>3. Puede comprobar que los botones de log-in y registro aparecen nuevamente.</li></ol>	
<b>Extensiones</b>	

### *Ver la sección de blog de una clase*

<b>Caso de uso</b>	Ver la sección de blog de una clase
<b>Actor principal</b>	Cliente potencial/Familiar de alumno
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Accede a la página web.</li><li>2. Selecciona la clase en la barra de navegación.</li><li>3. Todas las entradas que hacen referencia a esa clase aparecen en pantalla.</li></ol>	
<b>Extensiones</b>	

### *Ver una entrada del blog en detalle*

<b>Caso de uso</b>	Ver una entrada del blog en detalle
<b>Actor principal</b>	Cliente potencial/Familiar de alumno
<b>Precondición</b>	El usuario ha accedido a la sección de blog de alguna clase
<b>Escenario principal de éxito</b>	
1. Pulsar el botón “Veure més” de la entrada que se quiera ver. 2. La información sobre la entrada aparece en pantalla	
<b>Extensiones</b>	

### *Crear un comentario en una entrada*

<b>Caso de uso</b>	Crear un comentario en una entrada
<b>Actor principal</b>	Familiar de alumno
<b>Precondición</b>	1. El usuario posee una cuenta creada previamente y ha iniciado sesión en ella. 2. El usuario ha accedido a una entrada del blog de alguna clase.
<b>Escenario principal de éxito</b>	
1. Pulsar el botón “Afegir comentari” situado al final de la entrada en cuestión. 2. El formulario de creación de comentario aparece en pantalla. 3. Rellena el formulario de manera correcta. 4. Presiona el botón “Afegir comentari”. 5. Es re-direccionado a la entrada del blog.	
<b>Extensiones</b>	
3.a Los datos introducidos en el formulario no son correctos. 3.b Al pulsar “Afegir comentari” aparece un mensaje informando sobre los errores.	

### *Eliminar comentarios propios*

<b>Caso de uso</b>	Eliminar comentarios propios
<b>Actor principal</b>	Familiar de alumno
<b>Precondición</b>	1. El usuario posee una cuenta creada previamente y ha iniciado sesión en ella. 2. El usuario ha accedido a una entrada del blog de alguna clase. 3. El usuario tiene un comentario aprobado en la entrada del blog.
<b>Escenario principal de éxito</b>	
1. Pulsar el botón “Eliminar” situado en el comentario. 2. El comentario ya no aparece en pantalla y ha sido eliminado.	
<b>Extensiones</b>	

### *Contactar con el centro*

<b>Caso de uso</b>	Contactar con el centro
<b>Actor principal</b>	Cliente potencial/Familiar de alumno
<b>Precondición</b>	-
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Accede a la página web.</li><li>2. Selecciona “Contacte” en la barra de navegación.</li><li>3. El formulario de contacto aparece en pantalla.</li><li>4. Rellena el formulario de manera correcta.</li><li>5. Aparece en pantalla un aviso conforme el mensaje se ha enviado correctamente.</li><li>6. Un correo electrónico es enviado a la cuenta de correo del centro con la información que el usuario ha introducido en el formulario.</li></ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"><li>4.a Los datos introducidos en el formulario no son correctos.</li><li>4.b Al pulsar “Contacta!” aparece un mensaje informando sobre los errores</li></ol>	

### **Casos de uso de los empleados del centro**

#### *Iniciar sesión como usuario administrador*

<b>Caso de uso</b>	Iniciar sesión como usuario administrador
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	Una cuenta con permisos de administrador ha sido creada previamente
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Accede a la sección de administración de la página web.</li><li>2. El formulario de inicio de sesión aparece en pantalla.</li><li>3. Rellena el formulario de manera correcta.</li><li>4. Accede al panel de administración de la página web.</li></ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"><li>3.a Los datos introducidos en el formulario no son correctos.</li><li>3.b Al pulsar “Iniciar sessió” aparece un mensaje informando sobre los errores</li></ol>	

#### *Cerrar sesión como usuario administrador*

<b>Caso de uso</b>	Cerrar sesión como usuario administrador
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Una cuenta con permisos de administrador ha sido creada previamente</li><li>2. Se ha iniciado sesión en dicha cuenta y se ha accedido al panel de administración</li></ol>
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Presiona en el botón “Finalitza sessió” en la barra de navegación.</li><li>2. Se cierra la sesión actual.</li><li>3. Aparece en pantalla un aviso conforme la sesión se ha cerrado correctamente.</li></ol>	
<b>Extensiones</b>	

### *Crear un aviso o noticia*

<b>Caso de uso</b>	Crear un aviso o noticia
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Una cuenta con permisos de administrador ha sido creada previamente</li><li>2. Se ha iniciado sesión en dicha cuenta y se ha accedido al panel de administración</li></ol>
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Presiona en el botón “Afe gir” de la sección avisos del panel de administración.</li><li>2. Aparece el formulario de creación de avisos.</li><li>3. Rellena el formulario de manera correcta.</li><li>4. Es re-direccionado a la sección de avisos donde se puede ver que el nuevo aviso ha sido creado.</li></ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"><li>3.a Los datos introducidos en el formulario no son correctos.</li><li>3.b Al pulsar “Desar” aparece un mensaje informando sobre los errores</li></ol>	

### *Editar un aviso o noticia*

<b>Caso de uso</b>	Editar un aviso o noticia
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Una cuenta con permisos de administrador ha sido creada previamente</li><li>2. Se ha iniciado sesión en dicha cuenta y se ha accedido al panel de administración</li></ol>
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Presiona en el botón “Modificar” de la sección avisos del panel de administración.</li><li>2. Seleccionar que aviso se quiere editar.</li><li>3. Editar el aviso siempre y cuando se respeten las restricciones establecidas.</li><li>4. Es re-direccionado a la sección de avisos donde se puede ver que el aviso ha sido editado.</li></ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"><li>3.a Los datos editados en el formulario no respetan las restricciones.</li><li>3.b Al pulsar “Desar” aparece un mensaje informando sobre los errores</li></ol>	



### *Eliminar un aviso o noticia*

<b>Caso de uso</b>	Eliminar un aviso o noticia
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Una cuenta con permisos de administrador ha sido creada previamente</li><li>2. Se ha iniciado sesión en dicha cuenta y se ha accedido al panel de administración</li></ol>
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Presiona en el botón “Avisos” para acceder a la sección de avisos en el panel de administración.</li><li>2. Seleccionar que aviso se quiere eliminar.</li><li>3. Pulsar en el botón “Eliminar”.</li><li>4. Es re-direccionado a la página de eliminación de aviso.</li><li>5. Pulsar en el botón “Si, n’estic segur”.</li><li>6. Es re-direccionado a la sección de avisos donde se puede ver que el aviso ha sido eliminado.</li></ol>	
<b>Extensiones</b>	

### *Crear una entrada en el blog de alguna clase*

<b>Caso de uso</b>	Crear una entrada en el blog de alguna clase
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Una cuenta con permisos de administrador ha sido creada previamente</li><li>2. Se ha iniciado sesión en dicha cuenta y se ha accedido al panel de administración</li></ol>
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Presiona en el botón “Afegir” de la sección Blog posts del panel de administración.</li><li>2. Aparece el formulario de creación de posts.</li><li>3. Rellena el formulario de manera correcta, seleccionando la clase a la que pertenecerá la entrada del blog.</li><li>4. Es re-direccionado a la sección de Blog posts donde se puede ver que el nuevo post ha sido creado.</li></ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"><li>3.a Los datos introducidos en el formulario no son correctos.</li><li>3.b Al pulsar “Desar” aparece un mensaje informando sobre los errores</li></ol>	

### *Editar un aviso o noticia*

<b>Caso de uso</b>	Editar un aviso o noticia
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Una cuenta con permisos de administrador ha sido creada previamente</li><li>2. Se ha iniciado sesión en dicha cuenta y se ha accedido al panel de administración</li></ol>
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Presiona en el botón “Modificar” de la sección Blog posts del panel de administración.</li><li>2. Seleccionar que post se quiere editar.</li><li>3. Editar el post siempre y cuando se respeten las restricciones establecidas.</li><li>4. Es re-direccionado a la sección de Blog posts donde se puede ver que el post ha sido editado.</li></ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"><li>3.a Los datos editados en el formulario no respetan las restricciones.</li><li>3.b Al pulsar “Desar” aparece un mensaje informando sobre los errores</li></ol>	

### *Eliminar un aviso o noticia*

<b>Caso de uso</b>	Eliminar un aviso o noticia
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Una cuenta con permisos de administrador ha sido creada previamente</li><li>2. Se ha iniciado sesión en dicha cuenta y se ha accedido al panel de administración</li></ol>
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Presiona en el botón “Blog posts” para acceder a la sección de posts en el panel de administración.</li><li>2. Seleccionar que post se quiere eliminar.</li><li>3. Pulsar en el botón “Eliminar”.</li><li>4. Es re-direccionado a la página de eliminación de post.</li><li>5. Pulsar en el botón “Si, n’estic segur”.</li><li>6. Es re-direccionado a la sección de posts donde se puede ver que el post ha sido eliminado.</li></ol>	
<b>Extensiones</b>	

### *Verificar comentarios de usuarios*

<b>Caso de uso</b>	Verificar comentarios de usuarios
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Una cuenta con permisos de administrador ha sido creada previamente</li><li>2. Se ha iniciado sesión en dicha cuenta y se ha accedido al panel de administración</li></ol>
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Presiona en el botón “Comments” para acceder a la sección de comentarios en el panel de administración.</li><li>2. Presionar en el botón “No” en la opción de filtrar por estatus de verificación.</li><li>3. Seleccionar que comentario se quiere verificar.</li><li>4. Pulsar en el campo “Verificado” del comentario.</li><li>5. Comprobar que este campo está ahora seleccionado.</li><li>6. Pulsar “Desar” para guardar los cambios.</li><li>7. Es re-direccionado a la sección de comentarios donde se puede comprobar que el comentario está verificado.</li></ol>	
<b>Extensiones</b>	

### *Eliminar comentarios*

<b>Caso de uso</b>	Eliminar comentarios
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	<ol style="list-style-type: none"><li>1. Una cuenta con permisos de administrador ha sido creada previamente</li><li>2. Se ha iniciado sesión en dicha cuenta y se ha accedido al panel de administración</li></ol>
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"><li>1. Presiona en el botón “Comments” para acceder a la sección de comentarios en el panel de administración.</li><li>2. Seleccionar que comentario se quiere eliminar.</li><li>3. Pulsar en el botón “Eliminar”.</li><li>4. Es re-direccionado a la página de eliminación de comentario.</li><li>5. Pulsar en el botón “Si, n’estic segur”.</li><li>6. Es re-direccionado a la sección de comentarios donde se puede ver que el comentario ha sido eliminado.</li></ol>	
<b>Extensiones</b>	

### Recibir mensajes de contacto

<b>Caso de uso</b>	Recibir mensajes de contacto
<b>Actor principal</b>	Empleado del centro
<b>Precondición</b>	Algún usuario ha rellenado el formulario de contacto correctamente y lo ha enviado con éxito.
<b>Escenario principal de éxito</b>	
1. La cuenta de correo del centro recibe un nuevo correo electrónico. 2. El correo tiene como asunto el asunto que el usuario especificó en la web. 3. El correo electrónico contiene el nombre del usuario que lo ha enviado, su dirección de correo y la explicación del motivo de contacto.	
<b>Extensiones</b>	

## 4.3 Historias de usuario

En el punto 2.2 Requisitos generales se exponían, de manera general, las diferentes funcionalidades con las que el sistema deberá contar para así cumplir todos los objetivos propuestos. Para entender mejor qué supone cada una de estas funcionalidades para los 3 grupos de actores mencionados previamente podemos desglosarlos en historias de usuario.

Para clasificar en mayor medida las historias de usuario, podemos agruparlas en épicas, cantidades importantes de trabajo y funcionalidades desglosables en tareas más pequeñas, las historias. A su vez, las épicas se pueden agrupar en iniciativas que representan los objetivos globales del proyecto.

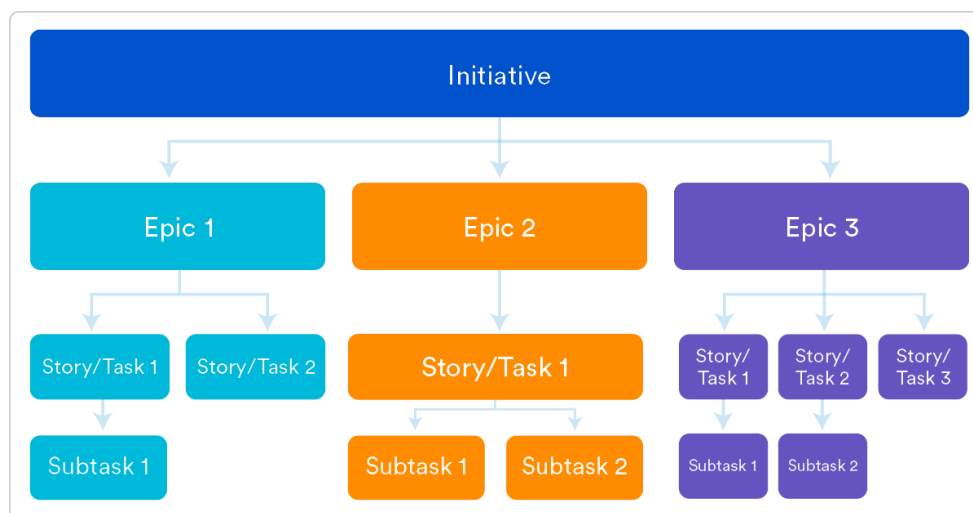


Ilustración 13: Estructura de la clasificación de historias de usuario.

En nuestro caso la iniciativa que engloba a todo es el hecho de implementar la aplicación. En cuanto a las historias de usuario, las podemos clasificar en 4 grandes pilares o épicas: Ofrecer información sobre el centro, sistema de usuarios tanto estándares como administradores, comunicar el día a día del centro y facilitar el contacto con él.

### 4.3.1 Historias de familiares de alumnos

#### Ofrecer información sobre el centro

Historia de usuario 1
Como familiar de un alumno,
Quiero <b>ver la información de contacto del centro</b> ,
Para poder contactar con él para comentar cualquier cuestión que se presente relacionada con el alumno.
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Al pulsar el botón de “contacte” en la barra de navegación se ha de direccionar a la página con la información de contacto.</li><li>• Se ha de mostrar la dirección del centro, el número teléfono y la dirección de correo electrónico.</li><li>• Esta información deberá ir acompañada de algún tipo de icono que la haga más identificable.</li><li>• Esta información deberá aparecer también en el pie de página de la página web y deberá estar siempre presente independientemente de la sección de la web en la que el usuario se encuentre.</li></ul>

Historia de usuario 2
Como familiar de un alumno,
Quiero <b>ver avisos o noticias importantes</b> ,
Para estar al día sobre el estado de la guardería.
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Las noticias y avisos deberán aparecer en la página principal.</li><li>• Para cada noticia o aviso se deberá mostrar su título, la fecha de creación y el cuerpo del mensaje.</li><li>• Se deberán ordenar los avisos y noticias por orden de relevancia, por tanto los mensajes más recientes serán los primeros en aparecer.</li></ul>

### Historia de usuario 3

Como familiar de un alumno,

Quiero **ver la información relacionada con la alimentación en el centro,**

Para saber que alimentos va a comer el alumno y poder actuar en caso de alergia, intolerancia o cualquier otra necesidad especial.

**Criterios de aceptación:**

- Al pulsar el botón de “cuina” en la barra de navegación se ha de direccionar a la página con la información sobre la alimentación.
- Se describirán los diferentes servicios que se ofrecen relacionados con la alimentación.
- Se mostrarán dos menús diferentes, uno para los bebés y otro menú de alimentos sólidos.
- Se mostrarán algunas imágenes de ejemplo de algunos platos que se cocinan en el centro.

### Historia de usuario 4

Como familiar de un alumno,

Quiero **ver la información relacionada con la alimentación en el centro,**

Para saber que alimentos va a comer el alumno y poder actuar en caso de alergia, intolerancia o cualquier otra necesidad especial.

**Criterios de aceptación:**

- Al pulsar el botón de “cuina” en la barra de navegación se ha de direccionar a la página con la información sobre la alimentación.
- Se describirán los diferentes servicios que se ofrecen relacionados con la alimentación.
- Se mostraran dos menús diferentes, uno para los bebés y otro menú de alimentos sólidos.
- Se mostraran algunas imágenes de ejemplo de algunos platos que se cocinan en el centro.

## Sistema de usuarios

### Historia de usuario 5

Como familiar de un alumno,

Quiero **poder darme de alta como usuario,**

Para poder comentar en el blog y así interactuar con la guardería.

#### **Criterios de aceptación:**

- Se debe poder acceder a la opción de registro desde todas las instancias de la aplicación.
- La opción de registro deberá ser accesible para todo el mundo.
- Se deberá poder especificar un nombre de usuario.
- Se deberá poder especificar un email.
- Se deberá poder especificar la contraseña. Será necesario especificarla 2 veces para evitar errores.
- Si el nombre de usuario ya está en uso, un mensaje informará de ello al usuario.
- Si las 2 contraseñas no coinciden, un mensaje informará de ello al usuario.
- Si la dirección de correo electrónico no es válida, un mensaje informará de ello al usuario.
- Si toda la información introducida es válida, se creará la cuenta y se avisará al usuario del éxito de la operación de registro.

### Historia de usuario 6

Como familiar de un alumno,

Quiero **poder iniciar sesión en mi cuenta,**

Para poder acceder a las funcionalidades que ello conlleva.

#### **Criterios de aceptación:**

- Se debe poder acceder a la opción de inicio de sesión desde todas las instancias de la aplicación.
- Se deberá poder especificar el nombre de usuario
- Se deberá poder especificar la contraseña.
- Si la combinación de nombre de usuario y contraseña no es correcta, un mensaje informará de ello al usuario.
- Si toda la información introducida es válida, se iniciará sesión en la cuenta especificada por el usuario y se direccionará a la página principal.

#### Historia de usuario 7

Como familiar de un alumno,

Quiero **poder cerrar la sesión de mi cuenta,**

Para evitar que terceras personas hagan uso de ella sin mi autorización.

**Criterios de aceptación:**

- Se debe poder acceder a la opción de cierre de sesión desde todas las instancias de la aplicación.
- Al pulsar en el botón de cerrar sesión, la sesión se cerrará.
- Al cerrar sesión, la opción de registrarse o iniciar sesión volverán a estar disponibles.

### Comunicar el día a día

#### Historia de usuario 8

Como familiar de un alumno,

Quiero **poder acceder a la sección de blog de la clase de mi hijo o hija,**

Para ver todas las entradas relacionadas con la clase y saber qué actividades han ocurrido.

**Criterios de aceptación:**

- Al pulsar el botón de alguna de las clases de la barra de navegación el usuario ha de ser direccionado a la sección de blog de dicha clase.
- En la sección de blog sólo deben aparecer entradas relacionadas con la clase seleccionada.
- Para cada entrada se deberá mostrar un título, una imagen de portada y su fecha de creación.
- Cada entrada deberá ir acompañada de un botón para saber más sobre ella.
- Se ordenarán las entradas por orden cronológico, las más recientes primero.
- En caso de haber muchas entradas, se dividirán en diferentes páginas.
- Cada página deberá contener como mínimo 5 entradas.
- Deberá ser posible navegar entre páginas de forma sencilla.

#### Historia de usuario 9

Como familiar de un alumno,

Quiero **poder ver una entrada del blog en detalle,**

Para ver y saber en detalle qué clase de actividades y juegos realiza mi hijo o hija en la guardería.

**Criterios de aceptación:**

- Al pulsar el botón de ver más de cualquiera de las entradas del blog el usuario deberá ser direccionado a la página de esa entrada en concreto.
- En la página de la entrada se mostrará el título, la fecha de creación, una explicación sobre la actividad si es necesario y hasta 10 imágenes que ilustren la actividad en cuestión.



- También se deberá mostrar una sección de comentarios.
- Para cada comentario de la sección de comentarios se mostrará el nombre de usuario del autor, la fecha de creación y el texto del comentario.
- En caso de no haber comentarios, un mensaje deberá aparecer informando de ello.
- Al final de la entrada deberá haber un botón que permita al usuario volver al blog de la clase.

## Facilitar el contacto

### Historia de usuario 10

Como familiar de un alumno,

Quiero **poder crear comentarios en las entradas del blog,**

Para poder dar mi opinión, exponer alguna duda o simplemente comentar sobre la actividad ilustrada en la entrada.

#### **Criterios de aceptación:**

- Para poder acceder a la funcionalidad de crear comentarios, el usuario deberá poseer una cuenta y haber iniciado sesión en ella.
- Si el usuario no ha iniciado sesión, en la sección de comentarios se le informará de la necesidad de hacerlo y en caso de hacer clic se le direccionará a la pantalla de inicio de sesión.
- Si el usuario ha iniciado sesión, la opción de añadir un comentario estará disponible. Si el usuario hace clic se le direccionará a la página de creación de comentario.
- Se permitirá al usuario introducir su comentario.
- Si el comentario no es válido o está vacío, un mensaje informará de ello al usuario.
- Se informará al usuario de la necesidad de la verificación del comentario antes de ser mostrado al público en la entrada del blog.
- Si el comentario introducido es correcto, el usuario será direccionado nuevamente a la entrada del blog y se registrara la fecha de creación del comentario y se le asignara el estado de no verificado.

### Historia de usuario 11

Como familiar de un alumno,

Quiero **poder eliminar mis comentarios,**

Para que dejen de ser mostrados públicamente en las entradas del blog.

#### **Criterios de aceptación:**

- Para poder eliminar un comentario, el usuario debe poseer una cuenta, tener como algún comentario verificado y haber iniciado sesión en su cuenta.
- Se mostrará un botón para eliminar el comentario sólo si es el autor.
- Tras pulsar el botón el comentario se eliminará y dejara de ser visible.

#### Historia de usuario 12

Como familiar de un alumno,

Quiero **poder contactar con el centro,**

Para poder preguntar o comentar con el centro cualquier duda o problema relacionado con el alumno.

##### **Criterios de aceptación:**

- El formulario de contacto debe estar disponible para todos los usuarios, independientemente de si han iniciado sesión o no.
- Al pulsar el botón de “contacte” en la barra de navegación se ha de direccionar a la página con el formulario de contacto.
- El formulario de contacto deberá permitir al usuario poner su nombre, su correo electrónico, el asunto de la consulta y su cuerpo.
- Si alguno de los datos introducidos no es válido, un mensaje informará de ello al usuario.
- Si los datos son válidos, al pulsar en el botón de contactar el formulario será procesado y se generara un correo electrónico con toda la información que será enviado al correo de la guardería de manera automática.
- Un mensaje deberá informar al usuario del éxito de la operación.

### 4.3.2 Historias de clientes potenciales

#### **Ofrecer información sobre el centro**

#### Historia de usuario 13

Como cliente potencial,

Quiero **ver la información de contacto del centro,**

Para poder contactar con él para comentar cualquier duda que tenga sobre el funcionamiento del centro o para obtener presupuestos/concertar una cita.

##### **Criterios de aceptación:**

- Al pulsar el botón de “contacte” en la barra de navegación se ha de direccionar a la página con la información de contacto.
- Se mostrará la dirección del centro, el teléfono y la dirección de correo electrónico.
- Esta información irá acompañada de algún tipo de icono que la haga más identificable.
- Esta información deberá aparecer también en el pie de página de la página web y estará siempre presente independientemente de la sección de la web en la que el usuario se encuentre.

#### Historia de usuario 14

Como cliente potencial,

Quiero **ver información relevante sobre el centro,**

Para poder hacerme una mejor idea sobre la guardería y para obtener información que me pueda ayudar a decidir si es la opción adecuada o no para mi familia.

**Criterios de aceptación:**

- La información más relevante sobre la guardería deberá aparecer en la página principal.
- Se proporcionará una breve descripción del centro.
- Se mencionarán diferentes puntos importantes sobre la guardería y se describirán con brevedad.
- La información irá acompañada de imágenes para ilustrar el contenido y hacerlo más visual.

#### Historia de usuario 15

Como cliente potencial,

Quiero **ver la ubicación del centro en un mapa,**

Para saber con exactitud dónde está ubicado.

**Criterios de aceptación:**

- Al pulsar el botón de “contacte” en la barra de navegación se ha de direccionar a la página con la información sobre la localización del centro.
- Se mostrará un mapa con la localización precisa de la guardería mediante el uso de Google Maps.
- Si se hace clic en mapa el usuario será direccionado a la página de Google Maps con la dirección del centro.

#### Historia de usuario 16

Como cliente potencial,

Quiero **ver las instalaciones del centro,**

Para conocer las diferentes estancias de la guardería y cómo están equipadas para satisfacer las diferentes necesidades de los niños.

**Criterios de aceptación:**

- Al pulsar el botón de “instalacions” en la barra de navegación se ha de direccionar a la página con la información sobre las instalaciones del centro.
- Para cada clase o sección de la guardería se ha de mostrar su nombre, una breve descripción sobre ella y una galería de imágenes.
- La galería de imágenes deberá mostrar las imágenes con un tamaño reducido.
- Al hacer clic en una imagen se deberá abrir una nueva ventana con un título de imagen, la propia imagen ampliada y un método de navegación para navegar entre las imágenes.

- Si se hace clic fuera de esta ventana la página deberá volverá a su estado original.

#### Historia de usuario 17

Como cliente potencial,

Quiero **ver los horarios de la guardería,**

Para saber si son compatibles con mis quehaceres personales.

##### **Criterios de aceptación:**

- Al pulsar el botón de “contacte” en la barra de navegación se ha de direccionar a la página con la información sobre los horarios.
- Se mostrarán los horarios de apertura del centro.
- Esta información deberá aparecer también en el pie de página de la página web y estará siempre presente independientemente de la sección de la web en la que el usuario se encuentre.

#### Historia de usuario 18

Como cliente potencial,

Quiero **ver la información relacionada con la alimentación en el centro,**

Para saber si se adaptan del todo a las necesidades alimentarias de mi hijo o hija.

##### **Criterios de aceptación:**

- Al pulsar el botón de “cuina” en la barra de navegación se ha de direccionar a la página con la información sobre la alimentación.
- Se describirán los diferentes servicios que se ofrecen relacionados con la alimentación.
- Se mostraran dos menús diferentes, uno para los bebés y otro menú de alimentos sólidos.
- Se mostraran algunas imágenes de ejemplo de algunos platos que se cocinan en el centro.

### Comunicar el día a día

#### Historia de usuario 19

Como cliente potencial,

Quiero **poder acceder a la sección de blog de las diferentes clases del centro,**

Para ver todas las entradas relacionadas con las diferentes clases del centro para ver las diferentes actividades y juegos que han ocurrido.

##### **Criterios de aceptación:**

- Al pulsar el botón de alguna de las clases de la barra de navegación el usuario ha de ser direccionado a la sección de blog de dicha clase.

- En la sección de blog sólo deben aparecer entradas relacionadas con la clase seleccionada.
- Para cada entrada se deberá mostrar un título, una imagen de portada y su fecha de creación.
- Cada entrada deberá ir acompañada de un botón para saber más sobre ella.
- Se ordenarán las entradas por orden cronológico, las más recientes primero.
- En caso de haber muchas entradas, se dividirán en diferentes páginas.
- Cada página deberá contener como mínimo 5 entradas.
- Deberá ser posible navegar entre páginas de forma sencilla.

#### Historia de usuario 20

Como cliente potencial,

Quiero **poder ver una entrada del blog en detalle,**

Para hacerme una mejor idea de la forma en la que trabajan las profesoras del centro y para ver las actividades y juegos en las que podría llegar a participar mi hijo o hija.

##### **Criterios de aceptación:**

- Al pulsar el botón de ver más de cualquiera de las entradas del blog el usuario deberá ser direccionado a la página de esa entrada en concreto.
- En la página de la entrada se mostrará el título, la fecha de creación, una explicación sobre la actividad si es necesario y hasta 10 imágenes que ilustren la actividad en cuestión.
- También se deberá mostrar una sección de comentarios.
- Para cada comentario de la sección de comentarios se mostrará el nombre de usuario del autor, la fecha de creación y el texto del comentario.
- En caso de no haber comentarios, un mensaje deberá aparecer informando de ello.
- Al final de la entrada deberá haber un botón que permita al usuario volver al blog de la clase.

## Facilitar el contacto

#### Historia de usuario 21

Como cliente potencial,

Quiero **poder contactar con el centro,**

Para poder comentar con el centro cualquier duda que tenga sobre el o para concertar una cita en persona en caso de querer más información.

##### **Criterios de aceptación:**

- El formulario de contacto debe estar disponible para todos los usuarios, independientemente de si han iniciado sesión o no.
- Al pulsar el botón de “contacte” en la barra de navegación se ha de direccionar a la página con el formulario de contacto.
- El formulario de contacto deberá permitir al usuario poner su nombre, su correo electrónico, el asunto de la consulta y su cuerpo.

- Si alguno de los datos introducidos no es válido, un mensaje informará de ello al usuario.
- Si los datos son válidos, al pulsar en el botón de contactar el formulario será procesado y se generará un correo electrónico con toda la información que será enviado al correo de la guardería de manera automática.
- Un mensaje deberá informar al usuario del éxito de la operación.

### 4.3.3 Historias de empleados del centro

#### Ofrecer información sobre el centro

##### Historia de usuario 22

Como empleado del centro,

Quiero **poder crear un aviso o noticia,**

Para poder comunicar a los padres información importante sobre el estado del centro.

##### **Criterios de aceptación:**

- Para poder crear un aviso o noticia, el profesor deberá haber iniciado sesión con una cuenta administradora en el panel de administración.
- Se deberá permitir introducir un título y un texto al que se le pueda dar formato.
- La fecha de creación deberá ser registrada automáticamente en el momento de crear la noticia.

##### Historia de usuario 23

Como empleado del centro,

Quiero **poder editar un aviso o noticia,**

Para poder corregir cualquier tipo de error que se haya cometido a la hora de crearla.

##### **Criterios de aceptación:**

- Para poder editar un aviso o noticia, el profesor deberá haber iniciado sesión con una cuenta administradora en el panel de administración.
- Se deberá permitir editar el título y el texto de la noticia.
- La fecha de creación no deberá ser editable.

##### Historia de usuario 24

Como empleado del centro,

Quiero **poder eliminar un aviso o noticia,**

Para dejar de mostrarla al público una vez no sea relevante o no sea necesaria.

##### **Criterios de aceptación:**

- Para poder eliminar un aviso o noticia, el profesor deberá haber iniciado sesión con una cuenta administradora en el panel de administración.

## Sistema de usuarios

### Historia de usuario 25

Como empleado del centro,

Quiero **poder iniciar sesión como usuario administrador**

Para poder administrar el contenido de la página web.

**Criterios de aceptación:**

- Para poder iniciar sesión como usuario administrador, una cuenta con permisos de administrador deberá haber sido creada previamente.

### Historia de usuario 26

Como empleado del centro,

Quiero **poder cerrar sesión como usuario administrador**

Para evitar que terceras personas hagan uso de ella sin mi autorización.

**Criterios de aceptación:**

- Al pulsar en el botón de cerrar sesión, la sesión se cerrará.
- Al cerrar sesión, la opción de iniciar sesión volverá a estar disponible.

## Comunicar el día a día

### Historia de usuario 27

Como empleado del centro,

Quiero **crear una entrada en el blog de alguna clase**

Para enseñar que actividad se ha realizado en clase.

**Criterios de aceptación:**

- Para poder crear una entrada en el blog, el profesor deberá haber iniciado sesión con una cuenta administradora en el panel de administración.
- Se deberá permitir introducir un título, un texto al que se le pueda dar formato, la clase a la que pertenece la entrada, una foto de portada y hasta 10 fotos.
- Al seleccionar la clase solo deberán ser posibles 3 opciones: bambis, cargols y ratolins.
- La fecha de creación deberá ser registrada automáticamente en el momento de crear la entrada.

#### Historia de usuario 28

Como empleado del centro,

Quiero **editar una entrada en el blog de alguna clase**

Para poder corregir cualquier tipo de error que se haya cometido a la hora de crearla.

**Criterios de aceptación:**

- Para poder editar una entrada, el profesor deberá haber iniciado sesión con una cuenta administradora en el panel de administración.
- Se deberá permitir editar el título, el texto de la entrada, la clase y las imágenes.
- La fecha de creación no deberá ser editable.

#### Historia de usuario 29

Como empleado del centro,

Quiero **poder eliminar un entrada del blog,**

Para dejar de mostrarla al público una vez no sea relevante o no sea necesaria.

**Criterios de aceptación:**

- Para poder eliminar una entrada, el profesor deberá haber iniciado sesión con una cuenta administradora en el panel de administración.

### Facilitar el contacto

#### Historia de usuario 30

Como empleado del centro,

Quiero **poder verificar comentarios de usuarios**

Para evitar que comentarios inapropiados sean publicados en las entradas de los blogs.

**Criterios de aceptación:**

- Para poder verificar un comentario, el profesor deberá haber iniciado sesión con una cuenta administradora en el panel de administración.
- Deberá ser posible cambiar el estado de un comentario de no verificado a verificado y viceversa.

#### Historia de usuario 31

Como empleado del centro,

Quiero **poder eliminar comentarios de usuarios**

Para evitar que se acumulen comentarios inapropiados que no son aptos para ser verificados.

**Criterios de aceptación:**

- Para poder eliminar un comentario, el profesor deberá haber iniciado sesión con una cuenta administradora en el panel de administración.



## Historia de usuario 32

Como empleado del centro,

Quiero **poder recibir mensajes de contacto**,

Para atender a las necesidades y dudas tanto de los padres que buscan una guardería como de los padres de alumnos del centro.

### **Criterios de aceptación:**

- Cuando un usuario rellene el formulario de contacto de manera correcta y lo envíe, un correo electrónico deberá ser enviado de manera automática a la cuenta de correo oficial de El Segon Cau.
- Este correo electrónico deberá ser visible en las principales plataformas de correo.
- El correo deberá contener el nombre del usuario, su correo y la explicación sobre el motivo de contacto. Como asunto el correo deberá tener el asunto que el usuario introdujo en el formulario.

## 4.4 Requisitos no funcionales

También es necesario tener en cuenta los requisitos no funcionales. Los requisitos no funcionales son todos aquellos requisitos que el proyecto debe cumplir pero que no están relacionados directamente con el funcionamiento de la aplicación. Para describirlos se utilizará la terminología Volere [7].

### Requisito de apariencia (*Volere 10a.*)

La apariencia de la aplicación debe ir acorde con la temática del proyecto y se debe ajustar a las expectativas de la organización.

#### **Justificación**

Uno de los puntos más importantes a la hora de captar la atención de las personas interesadas en la guardería es su apariencia, por ello la página deberá ser atractiva visualmente.

### Requisito de facilidad de uso (*Volere 11a.*)

La aplicación deberá ser sencilla de utilizar e intuitiva y ser lo más inclusiva posible.

Se deberá mostrar la información de manera simple y muy visual.

#### **Justificación**

La aplicación web será usada por un gran número de tipos diferentes de personas con diferentes niveles de conocimiento tecnológico. Para garantizar que todos puedan usarla de manera adecuada, es necesario que sea fácil de utilizar y muy intuitiva.

#### Requisito de facilidad de uso (*Volere 12a.*)

La aplicación deberá rendir a una velocidad aceptable que permita el pleno uso de las diversas funcionalidades sin problema.

##### **Justificación**

Los usuarios esperan una aplicación web eficaz y fiable. La satisfacción de los usuarios se reducirá si la aplicación se demora demasiado al hacer las tareas necesarias, siendo contraproducente para los intereses del centro. Por tanto, la web deberá ser rápida y eficiente.

#### Requisito de fiabilidad y disponibilidad (*Volere 12d.*)

Deberá ser posible acceder a la página web independientemente de la hora o el día. Idealmente la aplicación estará disponible las 24 horas del día, todos los días del año. En caso de fallo del sistema o avería, se deberá solucionar lo antes posible.

##### **Justificación**

Un usuario no quedará satisfecho si la página no está disponible cuando desee hacer uso de ella, por tanto deberá estar online el máximo de tiempo posible.

#### Requisito de producción (*Volere 13c.*)

La aplicación deberá ser accesible con facilidad y funcionar en los principales navegadores de manera gratuita.

##### **Justificación**

La aplicación web busca ser utilizada por el mayor número de personas posibles por tanto debemos facilitar al máximo el acceso a ella.

#### Requisito de privacidad y seguridad (*Volere 15c.*)

Solo se garantizará acceso a la información y a los datos a las personas con los permisos necesarios y no permitir el acceso a usuarios no autorizados.

##### **Justificación**

Al manejar datos de carácter personal como nombres y correos electrónicos es de vital importancia garantizar la seguridad de los datos y la confianza de los usuarios.

#### Requisito de legalidad (*Volere 17a.*)

La aplicación web no violará en ningún momento ninguna ley vigente de cualquier tipo, actuando siempre en el marco de la legalidad.

##### **Justificación**

Es importante cumplir con la ley para evitar posibles problemas de carácter legal como denuncias.

## 4.4 Modelo conceptual de datos

En este punto se describirán brevemente las clases del modelo conceptual junto con su diagrama UML.

### Usuario

Representa a todos los usuarios que harán uso de la aplicación web. Contiene información personal relacionada con el usuario como su nombre y contraseña. Podemos clasificar a los usuarios en 2 tipos diferentes:

- **Administrador:** Los usuarios administradores se corresponden con los empleados del centro. Tienen acceso al panel de administración para gestionar el contenido de la web.
- **Estándar:** Representan a los familiares de alumnos y a las personas interesadas en la guardería. Harán uso de la parte pública de la web.

### Permisos

Los permisos permiten a los usuarios gestionar datos e información específica de la aplicación. Por ejemplo, un permiso importante es la posibilidad de crear entradas en los blogs.

### Permisos Usuario

Dependiendo del tipo de usuario se tendrán unos permisos u otros.

### Grupo

Para gestionar permisos de manera más sencilla, se pueden crear grupos de usuarios con una lista de permisos pre-establecidos.

### Avisos

Representa los avisos y noticias importantes que el centro necesita comunicar a los usuarios.

### Consulta

Cualquier tipo de duda o cuestión que un usuario estándar pueda tener y que desee comunicar al centro.

### Post

Conjunto de texto e imágenes utilizado para representar una actividad o evento que haya ocurrido en clase.

### Clase

Representa las clases del centro. Diferenciamos 3 clases diferentes: Ratolins, Bambis y Cargols.

**BlogPost**

Cada una de las entradas de los blogs. Está compuesto por la información del post y por la clase en la que se llevó a cabo la actividad.

**Comentario**

Un texto breve usado para expresar alguna idea.

**Comentario Blog**

Representa cada uno de los comentarios de cada una de las entradas de los blogs. Tiene un campo específico que indica si el comentario es apto para mostrarse en la página web o no.

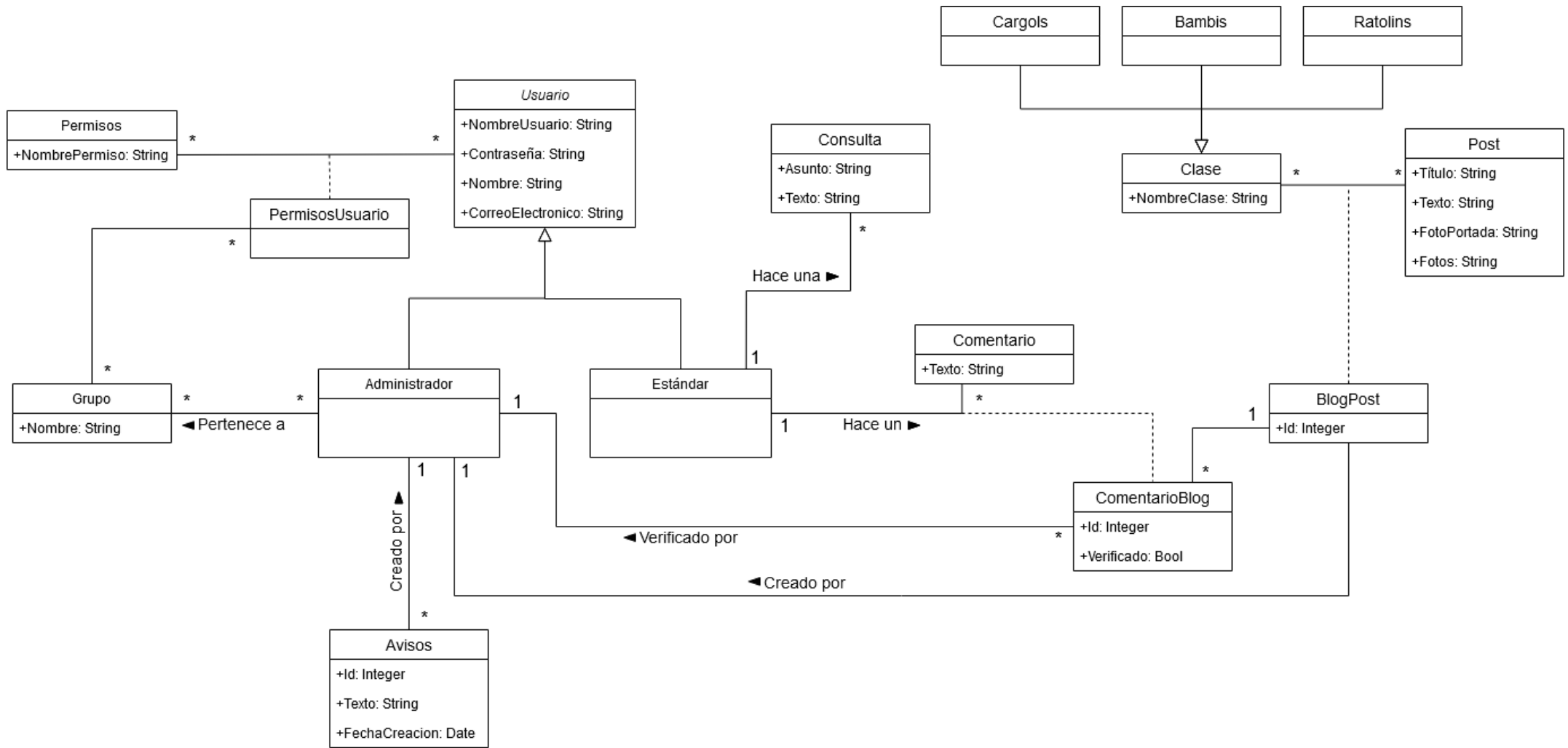


Ilustración 14: Diagrama UML del modelo conceptual de datos

## 4.4.1 Restricciones textuales de integridad

### 1. Claves primarias

(Usuario, NombreUsuario), (Permisos, NombrePermiso), (Grupo, Nombre), (Avisos, Id), (ComentarioBlog, Id), (BlogPost, Id), (Clase, NombreClase)

2. Un usuario administrador es a su vez un usuario estándar pero un usuario estándar no puede ser un usuario administrador.

3. Un usuario administrador no puede tener un permiso de usuario asociado a él si ya forma parte de un grupo que incluye dicho permiso.

4. No pueden haber dos grupos de usuarios diferentes con el mismo conjunto de permisos.

5. La clase asociada a un post debe estar relacionada con el contenido del post.

6. Un post puede no contener texto o imágenes.

8. Un post no puede no contener texto e imágenes a la vez.

9. Un comentario solo puede ser verificado si no lo estaba previamente.

## 5. Entorno de desarrollo

Una vez definidos y especificado el contenido de la aplicación web a desarrollar, podemos empezar a profundizar en las diferentes herramientas, sistemas y tecnología que se empleará para llevarlo a cabo.

A continuación se explicarán, de manera breve, algunos de los sistemas de desarrollo web que se tuvieron en consideración previa al desarrollo de la aplicación y por qué se ha optado por utilizar uno de ellos por encima de los demás. También se explicará que tecnología será utilizada durante el transcurso del desarrollo del proyecto.

### 5.1 Sistemas de desarrollo web

El área de desarrollo e implementación de aplicaciones web es bastante extenso y, por lo tanto, existen un gran número de herramientas y métodos para su creación. Antes de empezar el proyecto se analizaron principalmente 2 tipos de herramientas: Sistemas de gestión de contenidos y “web frameworks” o entornos de trabajo web.

#### 5.1.1 Sistemas de gestión de contenidos

Los sistemas de gestión de contenidos son sistemas software desarrollados para que cualquier usuario pueda administrar y gestionar los contenidos de una web con facilidad y sin conocimientos de programación web previos. De todos los SGCs disponibles pondremos especial interés en dos:

#### **Wordpress**

Creado en 2003, Wordpress [8] es la herramienta líder en creación de aplicaciones web y sistemas de gestión de contenidos. Se estima que cerca del 38.5% de todas las webs que hacen uso de este tipo de sistemas utilizan Wordpress [9]. Fue programado en PHP y MySQL y es de código libre, haciendo así posible la evolución del programa gracias a la colaboración de los usuarios. Uno de los puntos fuertes de Wordpress son los plugins o módulos, contenido creado por la comunidad que puede ser añadido a tu proyecto.



Ilustración 15: Logotipo de WordPress

## **Blogger**

Aunque es posible que no cuente con la relevancia que una vez tuvo, vale la pena mencionar Blogger, ya que es la herramienta que se utilizó para crear la página web actual de El Segon Cau. Blogger nació en el año 1999 [10] gracias a Pyra Labs como una de las primeras herramientas dedicadas enteramente a la publicación de blogs. Blogger permite crear blogs de manera sencilla y es ideal para aquellas personas o empresas que están empezando a promocionarse en internet. Esta sencillez tiene un precio y es que las diferentes opciones y herramientas que Blogger deja a nuestra disposición son bastante básicas y no muy potentes.



Ilustración 16: Logotipo de Blogger

### **5.1.2 Conclusiones**

Hay varios puntos en contra comunes entre los diferentes sistemas de gestión de contenidos. La gran mayoría se rigen por plantillas prefijadas creadas previamente, haciendo el diseño de la aplicación web algo más rígido. También es común el contenido “Premium” que, aun siendo útil, se desbloquea mediante pagos periódicos. Además, cada gestor de contenido se utiliza de manera única e independiente al resto, haciendo difícil aplicar los conocimientos adquiridos durante el grado. Por estas razones se decidió no optar por este tipo de creador de aplicaciones web y, por el contrario, utilizar un web framework para una máxima libertad a la hora de desarrollar la aplicación.

### **5.1.3 Web Frameworks**

Los frameworks web son un conjunto de herramientas, estilos y librerías dispuestas a través de una estructura o esqueleto base para el desarrollo de aplicaciones web muy escalables y sencillas de mantener. Para este proyecto se analizaron principalmente 2 frameworks:

#### **Ruby on Rails (RoR)**

Creado en 2003 y escrito en el lenguaje de programación Ruby, Ruby on Rails [11] es un framework web de código libre y de uso gratuito. Basa sus cimientos en el patrón Modelo-Vista-Controlador con el objetivo de evitar que al realizarse cambios en una parte del código, estos perjudiquen a otra parte no relacionada. Uno de los mayores atractivos y puntos fuertes de RoR son sus librerías o “gemas”. Las gemas son conjuntos de códigos creados por los propios usuarios que se pueden instalar en nuestro proyecto para ayudarnos a realizar funciones específicas con facilidad.





Ilustración 17: Logotipo de Ruby on Rails

## Django

Django [12] es un framework de aplicaciones web gratuito y de código abierto escrito en Python. Creado por la empresa Lawrence Journal-World, fue lanzado al público en 2005. Al igual que Ruby on Rails, Django sigue la filosofía del patrón Modelo-Vista-Controlador. Django incluye de serie un gran número de librerías que hace del desarrollo de aplicaciones web algo más sencillo y rápido. También cuenta con un panel de control para administrar nuestra aplicación de una manera más fácil e intuitiva.



Ilustración 18: Logotipo de Django.

### 5.1.4 Conclusiones

Tras analizar los pros y los contras de los 2 frameworks mencionados previamente, se optó por utilizar Django. El motivo principal que impulsó esta elección es el lenguaje de programación a utilizar. La sencillez de Python y su uso en varias de las asignaturas de la carrera como Criptografía o Ingeniería del Conocimiento y Sistemas Distribuidos Inteligentes hace que este lenguaje no sea algo ajeno al desarrollador a diferencia de Ruby, con el que no se tiene tanta experiencia y por lo tanto se debería emplear tiempo en aprender el lenguaje y no en implementar la aplicación. Además, Django permite administrar y evaluar todos los componentes del sistema de manera visual e intuitiva gracias al panel de administración, herramienta que jugará un papel muy importante a la hora de ir actualizando la aplicación web por parte de las empleadas del centro.

## 5.2 Tecnología y herramientas utilizadas

### Django

Como se mencionaba en el punto anterior, Django es el framework web que se utilizará para llevar a cabo el desarrollo e implementación de la aplicación web.

Su relativa sencillez a la hora de utilizar, la gran cantidad de información y documentación, su versatilidad, la existencia de un panel de administración entre otros hacen de Django la herramienta idónea para llevar a cabo este proyecto.

### Python

El hecho de escoger Django como herramienta de desarrollo implica el uso del lenguaje de programación Python.

Python aparece por primera vez al principio de los 90 de la mano de Guido Van Rossum, un ingeniero holandés que trabajaba por aquel entonces en el Centro de Investigación de Ciencias de la Computación holandés y que decidió crear un nuevo lenguaje de programación como un hobby.

Desde entonces Python no ha parado de estar en desarrollo, evolucionando y creciendo día a día. Actualmente cuenta con más de 50 versiones diferentes, siendo la 3.9.1 la más reciente.

Uno de los puntos principales en la filosofía de Python es el ofrecer una sintaxis de código simple y fácil de entender. Esto hace que Python sea un lenguaje sencillo, legible y elegante que atiende a un conjunto de reglas que hacen muy corta su curva de aprendizaje.

Una de las grandes ventajas de trabajar con Python es su comunidad. Al ser uno de los lenguajes de programación más populares, existe una gran cantidad de información, tutoriales y contenido en un proceso de crecimiento constante.

### HTML

HTML o *Hypertext Markup Language* es el lenguaje con el que se define el contenido de las páginas web. Podríamos pensar en HTML como un conjunto de etiquetas que sirven para definir los elementos que formarán una página web como el texto, imágenes, links...

Nacido en 1990 como una herramienta para procesar texto e imágenes en la web, ha ido evolucionando hasta permitir poder implementar casi todo lo que uno se imagine.

Al ser un lenguaje basado en etiquetas, su sintaxis y uso es muy diferente respecto a lenguajes más orientados al trato y cómputo de datos como Python, Java o C.

## CSS

CSS o *Cascading Style Sheets* es un lenguaje para el desarrollo web que se centra en estilizar los diferentes elementos que conforman una página. Desarrollado en 1996 por el *World Wide Web Consortium* para ofrecer una herramienta que permitiese dar formato a las páginas.

HTML y CSS comparten una relación muy fuerte. Con HTML se define el contenido de la página y con CSS se estiliza para así mejorar su apariencia.

## JavaScript

JavaScript es un lenguaje de programación o de secuencias de comandos que permite implementar funciones complejas en páginas web especialmente útil a la hora de animar y controlar contenido multimedia. En definitiva, con JavaScript podemos hacer que las páginas web sean mucho más interactivas.

## SQLite

SQLite es un sistema de gestión de base de datos relacional. SQLite está diseñado para que la base de datos, en vez de necesitar programas externos para su gestión, esté directamente integrada en la aplicación pasando a formar parte del sistema.

Esto es posible gracias a que la base de datos está contenida por completo en un solo fichero, desde donde el programa principal leerá y actualizará la información según se necesite. El hecho de tratarse de un solo fichero de poco peso al que es posible acceder de manera directa hace que las operaciones se realicen con una gran velocidad.

## GitHub

Como se comentó en el punto sobre la metodología de trabajo, GitHub es el entorno web que se utilizara para el control de las versiones del proyecto. Todos los ficheros necesarios para que el proyecto funcione serán alojados en un repositorio de GitHub que se irá actualizando según avance el desarrollo de la aplicación.

## Taiga

También mencionado previamente, Taiga es la herramienta escogida para la gestión de las tareas y del trabajo a realizar. Taiga nos permite dividir el proyecto por sprints, asignando historias de usuario a cada sprint y desglosando cada historia en tareas a realizar, facilitando el seguimiento del trabajo ya hecho o por hacer.

## PyCharm

PyCharm es un entorno de desarrollo integrado creado por la compañía JetBrains y especializado en la programación en Python. Gracias a un convenio con la UPC, los estudiantes pueden utilizar la versión *Professional* de manera gratuita que proporciona herramientas muy útiles como análisis de código, detección de errores y debugger, integración con sistemas de control de versiones como GitHub y lo más importante, permite el desarrollo web mediante el uso de Django.

## 6. Diseño

Siguiendo con la parte técnica del proyecto y una vez mencionadas las tecnologías a utilizar, podemos continuar con una explicación general del diseño del sistema. En este punto veremos qué supone trabajar con Django, sus partes, que arquitectura se utiliza y como se traduce dicha arquitectura al uso del patrón Model-Template-View. También veremos un ejemplo del diseño de una funcionalidad para entender mejor como funciona este patrón. Por último se verá el diseño de las estructuras de datos necesarias para implementar la aplicación.

### 6.1 Partes del sistema

#### Frontend

Proporciona la interactividad con el usuario y el apartado gráfico de la aplicación web. El objetivo del frontend es mostrar los datos e información enviados por el backend al usuario y comunicar los datos y peticiones introducidos por el usuario. Lo conforma el navegador que procesa los ficheros HTML, CSS y JS necesarios para mostrar las páginas.

#### Backend

Es la base de la aplicación web. Se encarga del procesado de datos, de la lógica interna de las diversas funcionalidades de la aplicación, se comunica con la base de datos para obtener o bien almacenar información y proporciona al frontend todo lo necesario para mostrar el contenido al usuario. Lo conforma el web framework django.

#### Base de datos

Se encarga de almacenar los datos de las diferentes partes y contenidos de la aplicación. Está compuesta por una base de datos SQLite3.

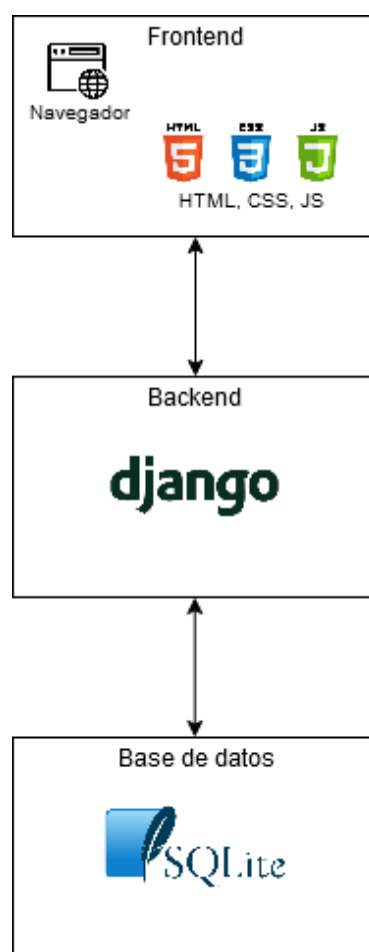


Ilustración 19: Partes del sistema

### 6.2 Arquitectura

Teniendo en cuenta la estructura que seguirá el sistema, la opción más óptima a seguir es una arquitectura por capas. El diseño por capas es un modelo de desarrollo de software cuyo objetivo primordial es el desacople o separación de las partes que componen el sistema. En nuestro caso contaremos con 3 capas diferentes: la capa de presentación, la capa de datos y la capa de dominio. El hecho de separar la aplicación en

capas independientes permite realizar cambios o añadir nuevo contenido en una de las capas sin la necesidad de cambiar nada en las otras.

## Capa de presentación

La capa de presentación es la capa que ve el usuario. Esta capa presenta el sistema al usuario, le comunica la información y le permite interactuar con el sistema cuando sea necesario. Al tratarse de la interfaz gráfica, deberá ser atractiva y fácil de usar. Una vez capturada la información del usuario, será la encargada de comunicársela a la capa de dominio.

## Capa de dominio

En la capa de dominio encontramos todas las funcionalidades de nuestra aplicación. En ella se procesará la información y las peticiones introducidas por el usuario y se enviarán las respuestas necesarias para atender las peticiones. Esta capa se comunica tanto con la capa de presentación, para recibir las peticiones del usuario y para enviar las respuestas, como con la capa de datos para almacenar o recuperar datos si es necesario.

## Capa de datos

Es donde residen los datos y todo lo relacionado con el acceso a ellos. Está formada por una base de datos y su sistema de gestión. En nuestro caso contaremos con una base de datos integrada SQLite.

## 6.3 Patrón Model-Template-View

Con el fin de seguir esta arquitectura basada en capas, Django hace uso del patrón Model-Template-View o MTV.

El patrón MTV es una variación del conocido patrón Modelo-Vista-Controlador o MVC. El objetivo de estos patrones es permitir la división de la aplicación en tres partes interconectadas entre sí.

El *model* o modelo contiene la parte asociada a las estructuras de datos. Es el encargado de procesar los datos y de la comunicación entre la base de datos y las *views* o vistas. Cuando una vista precise de datos almacenados en la base de datos, el modelo los obtendrá y se los transferirá a la vista en un formato apto para ser tratados. Ocurrirá lo mismo en sentido contrario, cuando sea necesario guardar nuevos datos la vista los transferirá al modelo que a su vez los almacenará en la BD.

Como aplicación web, podemos diferenciar 2 grandes partes en su ejecución. El cliente, que lo podríamos entender como el dispositivo que solicita información y el servidor, que responde a dichas solicitudes. En Django esta comunicación cliente-servidor se realiza mediante objetos de solicitud y respuesta, en concreto objetos HTTP request y HTTP response. La *view* o vista se encarga de obtener datos que se encuentran en la base de datos a través del modelo, procesarlos y tratarlos según convenga, formatearlos y agruparlos en un objeto HTTP response para así poder enviárselos al cliente.

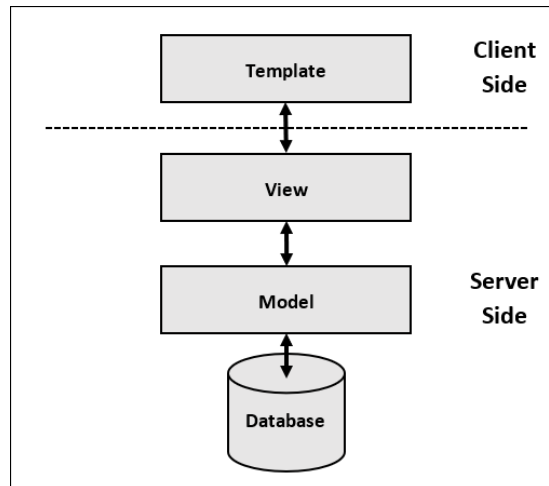


Ilustración 20: Estructura Cliente-Servidor

Por ultimo tenemos la *Template* o plantilla. Se corresponde con la capa de presentación mencionada previamente. Procesa todo lo relacionado con mostrar información al usuario como por ejemplo los ficheros HTML, CSS...

A modo de resumen, el *model* se encarga de acceder a los datos de la aplicación, la *view* procesa los datos y se los comunica al *template* y a su vez el *template* se los muestra al usuario. Podemos ver como hay un gran símil entre este patrón y las tres capas descritas en el punto anterior.

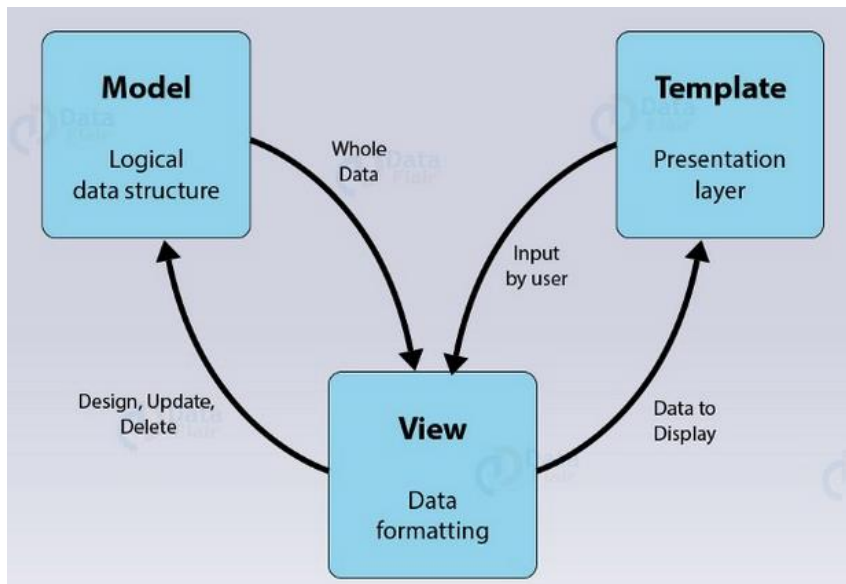


Ilustración 21: Diagrama del patrón MTV en Django.

## 6.4 Diseño de una funcionalidad con M-T-V

Tras explicar la idea y el funcionamiento del patrón Model-Template-View, podemos diseñar una funcionalidad a modo de ejemplo para esquematizar aún más las interacciones necesarias entre los diferentes componentes y clases. En este ejemplo en concreto representaremos la situación en la que un usuario decide acceder al blog de la clase de los Bambis.

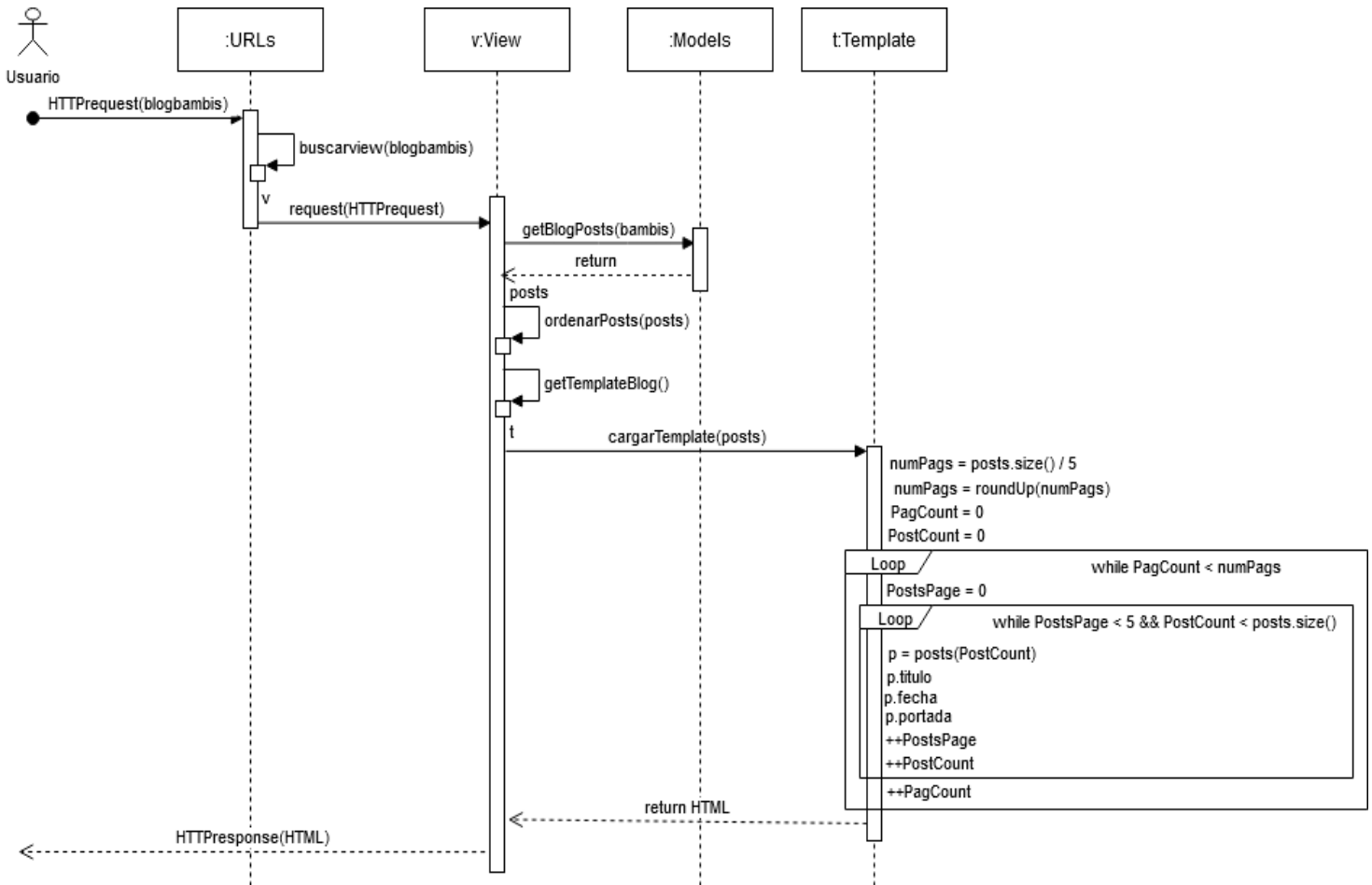


Ilustración 22: Diagrama de secuencia ver blog clase Bambis.

El diagrama empieza con el usuario accediendo a la dirección del blog. Esta acción envía una HTTP request que contiene la url del blog de los bambis. Esta url tiene asociada una view que podemos buscar. Una vez localizada la view le comunicamos la request. Como se mencionaba en el punto anterior, la view debe obtener los datos de los models para comunicárselos a la template. Los datos en cuestión son todos los posts que tienen asociada la clase de los Bambis. Una vez procesados pasamos los posts a la template correspondiente. La template divide los posts en páginas de 5 posts cada una y para cada post mostrará su título, fecha de creación y portada. El último paso es mostrar esta template, que no deja de ser un archivo .HTML, al usuario. Para ello enviamos un HTTP response con el HTML a mostrar, que será procesado por el navegador del usuario y mostrado por pantalla.

En este proceso podemos ver como interaccionan los tres pilares fundamentales en el desarrollo de aplicaciones con Django, las views, los models y las templates.

## 6.5 Diseño de los models

Como ya sabemos, los models son representaciones de las estructuras de datos que están presentes en la BD. Antes de empezar con la implementación de las diferentes funcionalidades es esencial diseñar los diferentes models que se utilizarán. Un buen diseño nos asegurará la disponibilidad de toda la información necesaria y nos evitará tener que realizar cambios estructurales a mitad de desarrollo. Contaremos con 6 models diferentes: Los usuarios, sus permisos, grupos de usuarios, avisos, BlogPosts y comentarios. Veremos cada uno de ellos en detalle en la sección de implementación.

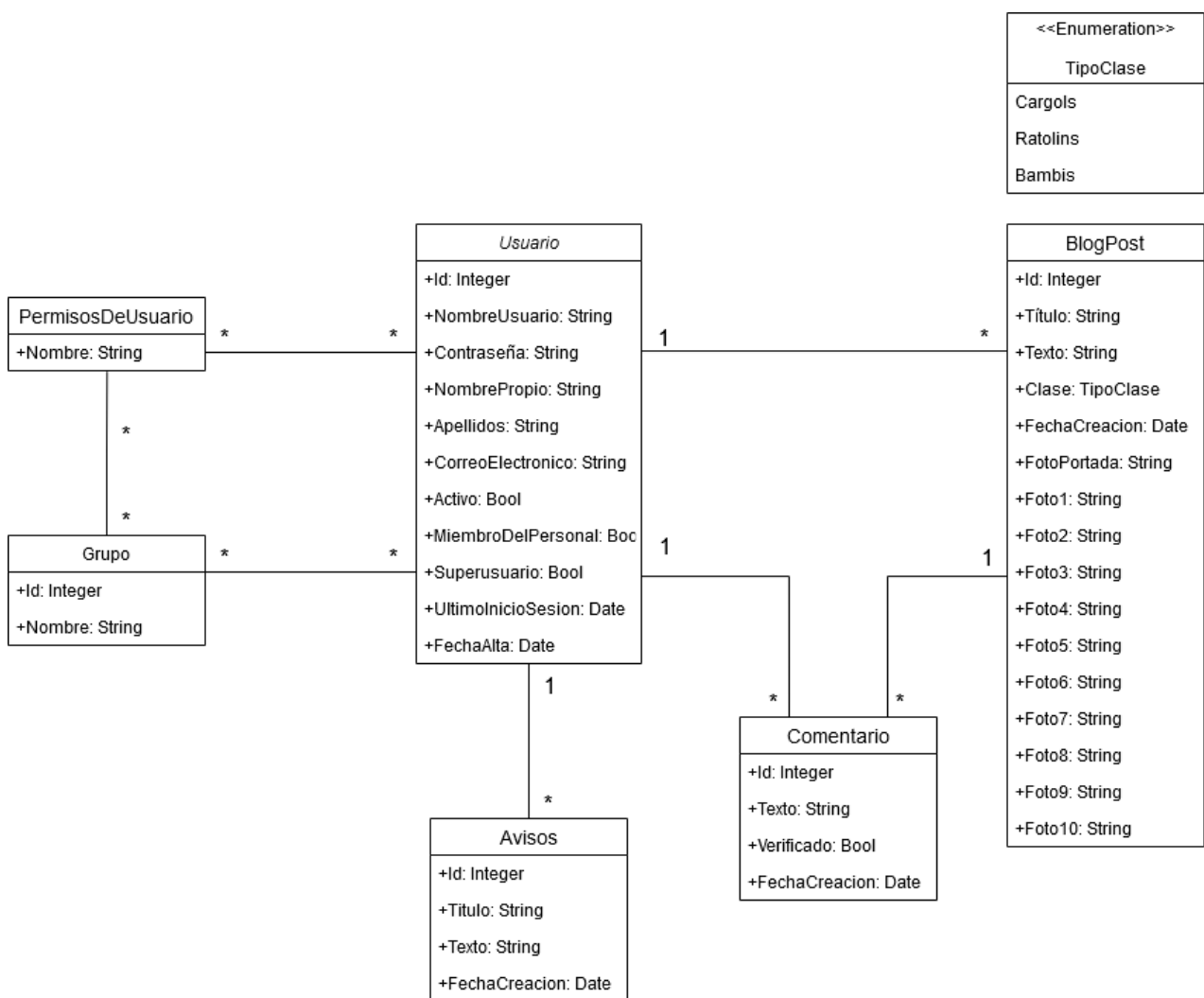


Ilustración 23: Esquema de los models.



# 7. Implementación

Una vez vista la arquitectura del sistema y cómo interactúan de manera general cada una de las capas entre sí podemos pasar a explicar con más detalle las partes que conforman la aplicación y como han sido implementadas.

## 7.1 Estructura del proyecto

Antes de empezar a profundizar en cada uno de los elementos que forman parte del proyecto, podemos echar un vistazo a la jerarquía de ficheros de la aplicación para hacernos una mejor idea sobre cómo está estructurado un proyecto desarrollado con Django.

Django ofrece la posibilidad de estructurar un proyecto tal y como el usuario desee, pero para este proyecto usaremos la estructura que viene dada por defecto ya que cambiarla no supone ninguna ventaja. La pieza fundamental de cualquier aplicación web desarrollada con Django es lo que llamamos el *Django Project* o proyecto Django, que engloba todas las configuraciones necesarias para coordinar y garantizar el buen funcionamiento de la aplicación. Los proyectos Django están a su vez formados por una o más *Apps*.

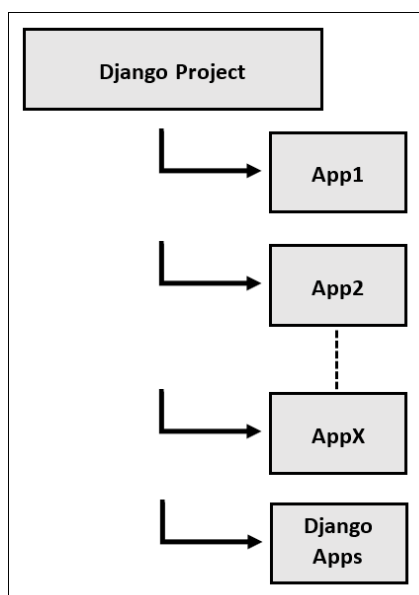


Ilustración 24: Estructura de un Django Project

Estas *Apps* son paquetes auto-contenidos que se deben especializar en una cosa en concreto. Por ejemplo, si nuestra aplicación web fuese una red social que además contase con una sección de compras, podríamos crear dos *Apps* diferentes, una para el aspecto de red social y otra para la tienda. El hecho de dividir los proyectos en *Apps* permite reutilizarlas en nuevos proyectos Django. En este proyecto en concreto desarrollaremos una única aplicación, la webapp de El Segon Cau.

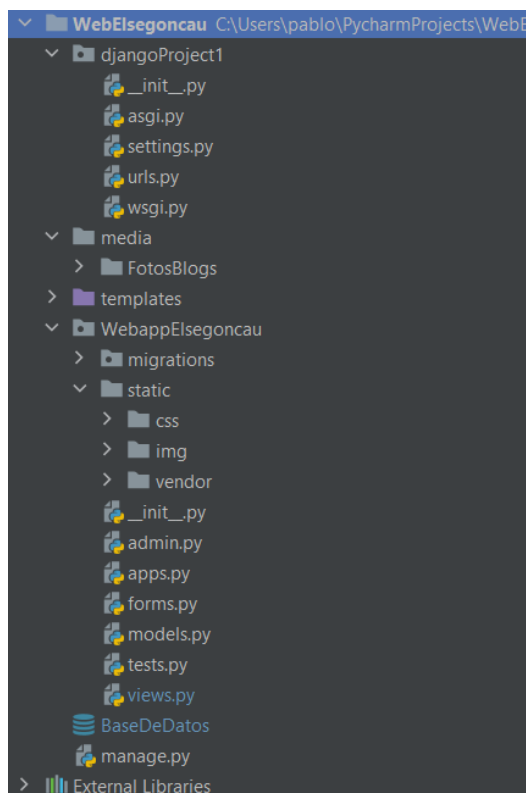


Ilustración 25: Estructura resumida del proyecto

Esta es la estructura resumida de nuestro proyecto Django. En esta imagen se muestran los archivos y carpetas más relevantes. Podemos ver tanto la carpeta de nuestro *Django Project* como la carpeta de nuestra aplicación tal y como se mencionaba previamente. A continuación se explicará de manera breve cada elemento.

### **djangoProject1**

- asgi & wsgi: Ficheros para la configuración de la aplicación una vez se quiera subir a un servidor.
- Settings: El fichero settings contiene toda la configuración del proyecto web. En él se declaran las Apps que formarán parte de la web, las rutas a los diferentes archivos y carpetas, que tipo de base de datos se va a utilizar, el lenguaje... entre muchas otras opciones.
- Urls: Contiene el esquema de las direcciones de cada una de las páginas que conformarán la aplicación web. Cada una de estas direcciones tendrá asociada su vista correspondiente.

### **media**

- FotosBlogs: En esta carpeta se almacenan todas las imágenes que hayan sido subidas como parte de una entrada de los blogs.

## **templates**

La carpeta templates contiene todos los ficheros HTML necesarios para poder mostrar al usuario cada una de las secciones de la web.

## **WebappElsegoncau**

- **Migrations:** Aquí encontramos las “migraciones”. Cada vez que se produce un cambio en la manera en que se representan los datos de la aplicación, por ejemplo crear una nueva tabla o eliminar algún campo en una ya existente, se crea un nuevo fichero de migración. Estos ficheros comunican a la base de datos que cambios se deben realizar.
- **Static:** La carpeta static contiene todos los ficheros necesarios para dar formato a nuestra web. Principalmente imágenes predefinidas que se mostrarán en la página, archivos CSS y archivos JavaScript.
- **Admin:** Fichero donde se configura la página de administración de Django. Podemos especificar que opciones queremos que aparezcan o que información se debe mostrar en ella.
- **Apps:** Contiene la configuración de la app. Podemos configurar diferentes atributos de la app como por ejemplo su nombre.
- **Forms:** En este archivo se definen todos los formularios que se utilizarán en la aplicación web.
- **Models:** En este archivo se definen todos los modelos de datos que se utilizarán en la aplicación web.
- **Tests:** Aquí se pueden crear tests mediante el módulo de Python unittest.
- **Views:** Todas las vistas que definirán el funcionamiento de la aplicación están definidas y programadas en este fichero.

## **BaseDeDatos**

Como su nombre indica, este fichero es nuestra base de datos y contiene toda la información relacionada con los usuarios y sus grupos, el blog, los comentarios y los avisos.

## 7.2 Models

Una vez vista la estructura del proyecto nos podemos centrar en describir los elementos más importantes que hacen posible la aplicación web. En primer lugar se especificarán los *models* o modelos.

En Django, un model es una clase de datos que actúa como puente entre la base de datos y el servidor. Podemos ver los models como representaciones de las estructuras de datos que serán utilizadas por la web y que estarán relacionadas directamente con la BD. Gracias a los models podemos implementar estas estructuras usando Python y, en consecuencia, podemos contar con una base de datos plenamente funcional sin la necesidad de introducir ni una sola línea de código SQL.

En nuestro proyecto contaremos con 6 models diferentes. Django nos ofrece 3 de ellos por defecto, los usuarios, los grupos de usuarios y los permisos. Los otros 3 models restantes estarán implementados desde cero y serán las entradas de los blogs o BlogPosts, los comentarios y los avisos.

### 7.2.1 Usuarios

Para hacer la vida más sencilla al desarrollador, Django deja a nuestra disposición un model para usuarios ya implementado. Este model se podría modificar o expandir en caso de ser necesario pero para nuestra aplicación el contenido por defecto es más que suficiente. A continuación veremos qué campos de información conforman un usuario.

- **NombreUsuario:** Cadena de como máximo 150 caracteres. Campo obligatorio. Como su nombre indica representa el nombre con el que un usuario desea representarse dentro de la aplicación.
- **NombrePropio:** Cadena de como máximo 150 caracteres. Campo opcional. El nombre real del usuario.
- **Apellidos:** Cadena de como máximo 150 caracteres. Campo opcional. El apellido del usuario.
- **CorreoElectronico:** Cadena de caracteres de tipo Email. Campo opcional. Si la información introducida no se corresponde con una dirección de email no se dará como válida.
- **Contraseña:** La contraseña encriptada del usuario. Campo obligatorio. Django no almacena la contraseña sin encriptar del usuario para garantizar la seguridad de la información.
- **Activo:** Es un booleano. Designa si la cuenta de usuario debe ser tratada como activa o no. Podemos marcar un usuario como inactivo en vez de eliminarlo por completo, de esta manera si algún otro model tiene a los usuarios como clave externa éstas no se romperán.

- **MiembroDelPersonal:** Es un booleano. Designa si la cuenta pertenece a un miembro del personal y por tanto puede acceder al panel de administración.
- **Superusuario:** Es un booleano. Dota a la cuenta de todos los permisos existentes.
- **UltimoInicioSesion:** Campo de tipo DateTime. Registra la fecha y hora del último inicio de sesión del usuario.
- **FechaAlta:** Campo de tipo DateTime. Registra la fecha y hora en la que se creó la cuenta.

Adicionalmente Django cuenta con 2 models más relacionados con los usuarios, los grupos y los permisos. El objetivo de los grupos es asignar un conjunto de permisos a varios usuarios a la vez y el objetivo de los permisos es dotar a un usuario con la capacidad de manipular información relacionada con los models.

## 7.2.2 Aviso

Uno de los models implementados es el Aviso. Como ya se ha mencionado en varias ocasiones, el centro necesitaba de una herramienta para comunicar las posibles noticias o información importante a los padres de los alumnos del centro. Para ello, se diseñó este model.

```
class Aviso(models.Model):
    titulo = models.CharField(max_length=150)
    texto = RichTextField(blank=False, null=False)
    fecha_creacion = models.DateTimeField(auto_now_add=True)

    class Meta:
        ordering = ['-fecha_creacion']

    def __str__(self):
        return self.titulo
```

Ilustración 26: Implementación del model Aviso.

A continuación se explicará el contenido de la imagen que se corresponde con la implementación del model mediante el uso de Python. Primero se definen los campos que conformarán un Aviso:

- **Título:** Cadena de como máximo 150 caracteres. Campo obligatorio. Representa el título que tendrá el aviso o la noticia. Un título debe ser relativamente breve por ello tiene un límite de tamaño.

- **Texto:** Es el cuerpo del aviso o la noticia y por tanto no tiene límite de caracteres. Campo obligatorio. El tipo del campo es RichTextField. Este tipo no viene por defecto con Django sino que está importado de una librería. RichTextField es un campo de texto con una interfaz para darle formato integrada. Esto es especialmente útil a la hora de introducir textos con varias partes, títulos, listas...

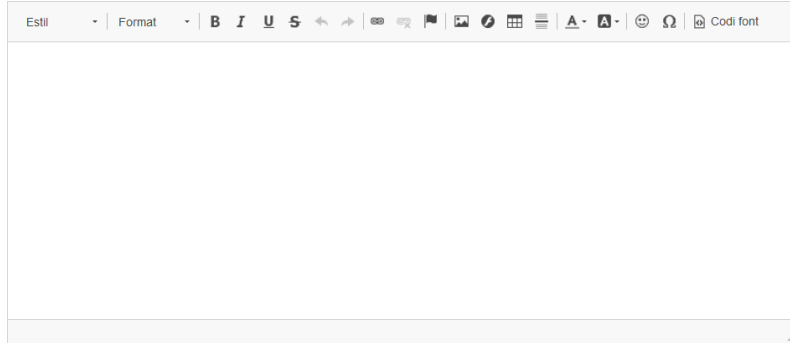


Ilustración 27: Campo de tipo RichTextField.

- **FechaCreación:** Campo de tipo DateTime. Registra la fecha y hora en la que se creó el aviso. Será añadido de manera automática en cuanto el usuario presione el botón de crear.

A parte de los campos, también está definido como queremos que se ordenen los avisos. Al estar fuertemente ligados con el tiempo, los avisos estarán ordenados de manera descendente según la fecha de creación. Esto quiere decir que una vez los mostremos en la web, los más recientes aparecerán primero.

### 7.2.3 Blog Post

```
class BlogPost(models.Model):

    class TipoClase(models.TextChoices):
        BAMBIS = 'BAMBIS'
        CARGOLS = 'CARGOLS'
        RATOLINS = 'RATOLINS'

    titulo = models.CharField(max_length=150)
    texto = RichTextField(blank=True, null=True)
    clase = models.CharField(max_length=15, choices=TipoClase.choices)
    fecha_creacion = models.DateTimeField(auto_now_add=True)
    foto_portada = models.ImageField(upload_to='FotosBlogs')
    foto1 = models.ImageField(blank=True, upload_to='FotosBlogs')
    foto2 = models.ImageField(blank=True, upload_to='FotosBlogs')
    foto3 = models.ImageField(blank=True, upload_to='FotosBlogs')
    foto4 = models.ImageField(blank=True, upload_to='FotosBlogs')
    foto5 = models.ImageField(blank=True, upload_to='FotosBlogs')
    foto6 = models.ImageField(blank=True, upload_to='FotosBlogs')
    foto7 = models.ImageField(blank=True, upload_to='FotosBlogs')
    foto8 = models.ImageField(blank=True, upload_to='FotosBlogs')
    foto9 = models.ImageField(blank=True, upload_to='FotosBlogs')
    foto10 = models.ImageField(blank=True, upload_to='FotosBlogs')
```

Ilustración 28: Implementación del model Blog Post.

Otro de los modelos implementados es el Blog Post. Los Blog Posts son cada una de las entradas de los blogs de las clases del centro y sirven para mostrar a los usuarios las diferentes actividades y eventos que realizan los niños en la guardería durante el curso. Como una imagen vale más que mil palabras, los Blog Posts cuentan con una gran capacidad para mostrar fotografías. A continuación se explicarán los elementos que forman un Blog Post:

- **TipoClase:** Es una subclase que se encuentra dentro de los Blog Posts. Está compuesta por 3 cadenas de caracteres, una para cada clase del centro. Es el equivalente a una *enumeration* en lenguaje UML.
- **Título:** Cadena de como máximo 150 caracteres. Campo obligatorio. Representa el título que tendrá la entrada del blog.
- **Texto:** Es el cuerpo de la entrada del blog y por tanto no tiene límite de caracteres. Este campo es opcional, una entrada del blog puede estar compuesta solo por fotos. El tipo del campo es RichTextField cuya explicación hemos visto previamente.
- **Clase:** Representa la clase del centro asociada al post. Es una cadena de caracteres y siempre será una de las 3 clases definidas en TipoClase. Es un campo obligatorio.
- **FechaCreación:** Campo de tipo DateTime. Registra la fecha y hora en la que se creó la entrada del blog. Será añadido de manera automática en cuanto el usuario presione el botón de crear.
- **FotoPortada:** Es la imagen de portada que ilustra el contenido de la entrada. Es un campo obligatorio.
- **Foto 1...10:** Cada entrada del blog permite hasta 10 imágenes diferentes para ilustrar la actividad o evento. Como no se sabe a ciencia cierta el número de imágenes que va a tener un post, todos los campos están marcados como opcionales.

## 7.2.4 Comentario

El último modelo implementado es el comentario. Los comentarios son pequeños textos que permiten a los usuarios expresar las posibles ideas que tengan sobre un Blog Post en concreto. A parte de contener la información sobre el comentario, también referenciará al usuario que lo ha realizado y al Blog Post donde se encuentra.

```
class Comment(models.Model):
    post = models.ForeignKey(BlogPost, related_name="comments", on_delete=models.CASCADE)
    autor = models.ForeignKey(User, on_delete=models.CASCADE)
    Comentari = models.TextField()
    fecha_creacion = models.DateTimeField(auto_now_add=True)
    verificado = models.BooleanField(default=False)

    class Meta:
        ordering = ['-fecha_creacion']

    def __str__(self):
        return self.post.titulo
```

Ilustración 29: Implementación del modelo Comentario.

- **Post:** Representa al Blog Post sobre el cual se comenta. Es una clave externa y será registrado automáticamente una vez se cree el comentario. Si un Blog Post es eliminado, todos los comentarios asociados a ese post serán eliminados también.
- **Autor:** Representa al usuario que realiza el comentario. Es una clave externa y será registrado automáticamente una vez se cree el comentario. Si un usuario es eliminado, todos sus comentarios serán eliminados también.
- **Comentari:** Es el comentario propiamente dicho y, por tanto, es obligatorio.
- **FechaCreación:** Campo de tipo DateTime. Registra la fecha y hora en la que se creó el comentario. Será añadido de manera automática en cuanto el usuario presione el botón de crear.
- **Verificado:** Es un booleano y por defecto estará asignado como False tras crear el comentario. El campo verificado dictamina si un comentario es apto para ser mostrado al público en la entrada del blog o no. Este proceso de verificación ocurrirá de manera manual a través del panel de administración y por parte de los empleados del centro.

Ordenaremos los comentarios de manera descendiente según la fecha de creación, de este modo los comentarios más recientes aparecerán los primeros.



## 7.3 Views & URLs

En Django, las vistas actúan de intermediarias entre los models y las templates. Obtendrán la información de la base de datos o de una fuente de datos externa, la procesarán y la entregarán a la plantilla que deba mostrarse al usuario en ese momento.

La vista decide qué datos se envían a la plantilla, ya sea actuando en base a las acciones que realice el usuario o en respuesta a otra lógica o procesos internos. Cada vista de Django realiza una función específica y tiene una plantilla asociada. A continuación podemos ver un pequeño diagrama que ilustra este proceso.

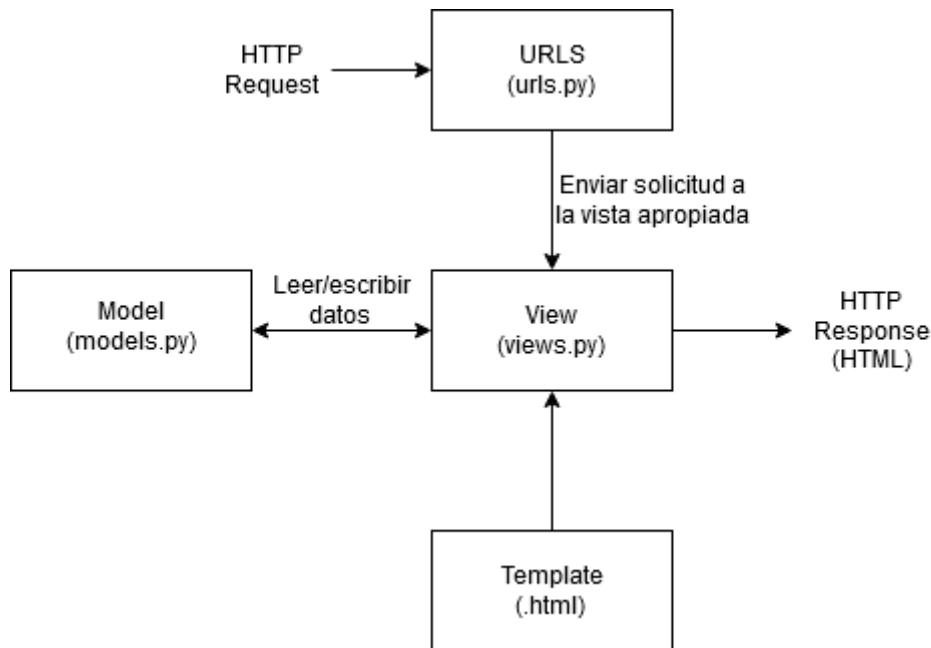


Ilustración 30: Como se relacionan los diferentes elementos de Django.

Otro aspecto importante a tener en cuenta son las URLs. Las URLs son las diferentes direcciones web que conforman nuestra aplicación. Una dirección, por si sola, es inútil. Para que tenga utilidad debe tener asociada toda la información que queramos mostrar al usuario. Es ahí donde entran en juego las views. Para cada URL, tendremos asociada su view correspondiente. Cuando un usuario acceda a una de las direcciones de la web, su view será la encargada de obtener toda la información necesaria y además decidir que plantilla (fichero HTML) se deberá cargar y mostrar al usuario.

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', home, name='home'),

    # usuarios
    path('register', registerPage, name='register'),
    path('login', loginPage, name='login'),
    path('logout', logoutUser, name='logout'),

    # paginasinformativas
    path('instalacions', instalacions, name='instalacions'),
    path('contacte', contacte, name='contacte'),
    path('cuina', cuina, name='cuina'),

    # blog
    path('classeratulins', BlogClasseRatulins.as_view(), name='classeratulins'),
    path('classebambis', BlogClasseBambis.as_view(), name='classebambis'),
    path('classecargols', BlogClasseCargols.as_view(), name='classecargols'),
    path('blogpost/<int:pk>', EntradaBlog.as_view(), name='blogpost'),

    # comentarios
    path('blogpost/<int:pk>/comentari', CrearComentario.as_view(), name='crearcomentario'),
    path('blogpost/<int:BlogPost_pk>/deletecomentari/<int:pk>', BorrarComentario.as_view(), name='borrarcomentario'),
]
```

Ilustración 31: Asociaciones URL-View.

En esta imagen podemos ver los diferentes grupos de direcciones de la aplicación web. Tenemos las 2 URLs principales, admin y home, las que tratan el tema de los usuarios, las páginas que dan información sobre el centro, las relacionadas con el blog y las relacionadas con los comentarios. Para cada una de ellas tenemos asociada su view correspondiente. Por ejemplo en `path('login', loginPage, name='login')`, la dirección se especifica primero, en este caso es `login`, seguida de su view `loginPage` y por último el nombre con el que nos referiremos a esta dirección en caso de necesitarla a la hora de programar. A continuación se mencionará cada una de las views implementadas en la aplicación y cuál es su función.

## Home

La view home representa la página principal de la web. Como en la página principal se muestra los avisos, deberá obtener todos los avisos existentes en la BD y adjuntárselos al fichero HTML que se mostrará al usuario.

## registerPage

Se encarga del registro de usuarios. Si el usuario ya ha iniciado sesión, se le redireccionará a la página principal, si no lo está, se le presentará el formulario de registro. Una vez haya rellenado el formulario, verificamos que los datos introducidos sean correctos. Si lo son crearemos un nuevo usuario guardándolo en la BD.

## loginPage

Se encarga del inicio de sesión de usuarios. Si el usuario ya ha iniciado sesión, se le re-direccionará a la página principal, si no lo está, se le presentará el formulario de login. Una vez el usuario haya rellenado el formulario, comprobamos si existe un usuario con esa combinación de nombre de usuario y contraseña. Si existe se inicia sesión y si no existe se muestra un mensaje de error.

## **logoutUser**

En caso de haber iniciado sesión, la cierra. Si no se ha iniciado sesión no hace nada.

## **Instalacions**

Manda renderizar la plantilla sobre las instalaciones del centro.

## **Contacte**

Manda renderizar la plantilla sobre la información de contacto que a su vez contiene el formulario de contacto. Una vez el usuario haya rellenado el formulario, se obtendrá la información introducida en él para crear el mensaje de correo electrónico que se enviará a la cuenta de correo de la guardería.

## **Cuina**

Manda renderizar la plantilla sobre la información relacionada con la alimentación en el centro.

## **BlogClasseRatolins, BlogClasseBambis, BlogClasseCargols**

Son las secciones de blog de cada una de las clases del centro. Dependiendo de la clase a la que haga referencia, se hará una búsqueda en la base de datos de los posts que tengan asociada dicha clase y luego se ordenaran en orden descendiente según su fecha de creación para que el más reciente aparezca primero. También se organizará el sistema de paginación de posts, permitiendo hasta 5 posts por página.

## **EntradaBlog**

Representa una entrada de un blog. Al tener la id del post en la propia URL, podemos acceder a la base de datos y obtener toda la información sobre él de manera directa. Esta información será adjuntada al fichero HTML para ser mostrada al usuario.

## **CrearComentario**

Se encarga de la creación de comentarios y solo estará disponible si el usuario ha iniciado sesión. Una vez el usuario haya rellenado el formulario de creación de comentario, se obtendrá la información tanto del usuario como de la entrada del blog y se registrará el nuevo comentario en la base de datos.

## **BorrarComentario**

Se encarga de eliminar comentarios. Cuando el usuario presione el botón de eliminar en uno de sus comentarios, se obtendrá su id. Con esta id lo podemos eliminar con facilidad de la BD.

## 7.3 Templates

Las plantillas de Django son una combinación de diseño HTML y sintaxis de Django, que es esencialmente código Python. Ambos componentes ayudan a generar páginas HTML de forma dinámica.

El propósito principal de las plantillas de Django es separar la representación de los datos con los datos en sí. Eso significa que el trabajo del navegador es renderizar el HTML que le envía el servidor mientras que Django proporciona todos los datos relevantes a la plantilla. Esto hace que el proceso sea mucho más fácil y las páginas se procesen fácilmente, ya que hay menos desorden tanto en el front-end como en el back-end.

Para este proyecto se han diseñado e implementado un conjunto de plantillas que juntas forman la parte de la aplicación web con la que interactúa el usuario.

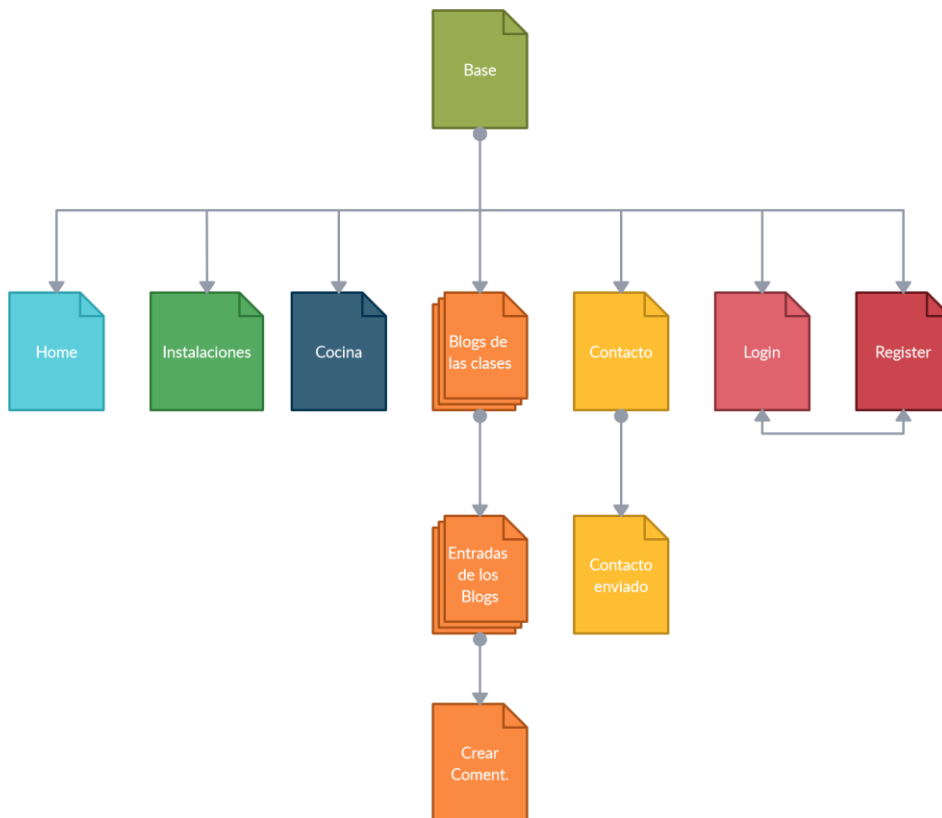


Ilustración 32: Jerarquía de templates

En este diagrama podemos ver la relación entre las diferentes templates del proyecto. A partir de la template Base, podemos ver como surgen todas las otras templates de la aplicación. Esto es debido a una práctica de programación de templates llamada herencia de plantillas.

La herencia de plantillas busca reducir el código redundante o duplicado a la hora de implementarlas. Es común que todas las plantillas de una misma página web sigan una estructura similar para mantener la temática y el estilo uniformes. Por ejemplo, es probable que todas las páginas de una web tengan el mismo menú de navegación o que todas tengan la misma cabecera. Si no se hiciese uso de la herencia, el código de cada uno de estos elementos se repetiría cada vez que se crease una plantilla nueva. Gracias a esta práctica podemos hacer que las plantillas “hereden” el código de la plantilla base que los contiene, evitando así la repetición innecesaria de código.

El diseño es el aspecto más importante a la hora de implementar templates, pues es lo que verá y con lo que interactuará el usuario cuando visite la página. A continuación definiremos la visualización de las templates más importantes mediante el uso de mock ups para hacernos una mejor idea del diseño de la aplicación.

## Base

Como acabamos de explicar, nuestra plantilla base contendrá todos aquellos elementos comunes que queramos que se repitan en las diversas páginas de nuestra web. En nuestro caso estos elementos serán el *header* o cabecera y el *footer* o pie de página. La cabecera contendrá el nombre del centro El Segon Cau y un menú de navegación para poder desplazarnos de una página a otra. El pie de página contendrá información relevante sobre la guardería como sus horarios, la información de contacto o algunos links de interés.

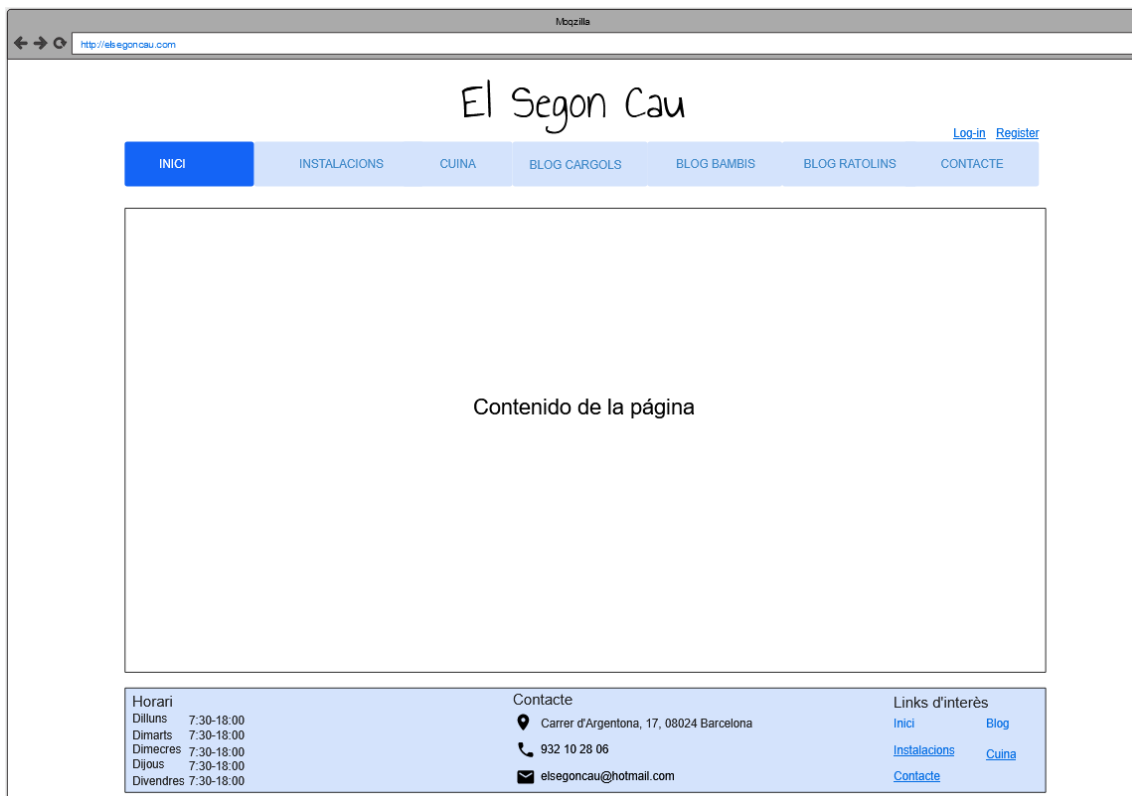


Ilustración 33: Mock up de la plantilla base.

Como cada una de las templates que no sea la template base “heredará” tanto la cabecera como el pie de página, solo será necesario implementar en ellas el contenido que aparecerá en la sección “Contenido de la página” de la imagen.

## Home

La template home es la página principal y la primera página que verá el usuario al acceder a nuestra web. El propósito de esta página es informar sobre el centro y a su vez captar la atención del usuario, por tanto se alternarán tanto texto como imágenes para hacer la experiencia muy visual. Al principio encontraremos una breve introducción seguido de unas tarjetas de información que explican aspectos importantes sobre el centro. Cada una de estas tarjetas va acompañada de una imagen relacionada para ilustrar el concepto. Por último encontramos la sección de avisos y noticias donde se informará al usuario de posibles acontecimientos o información importante.



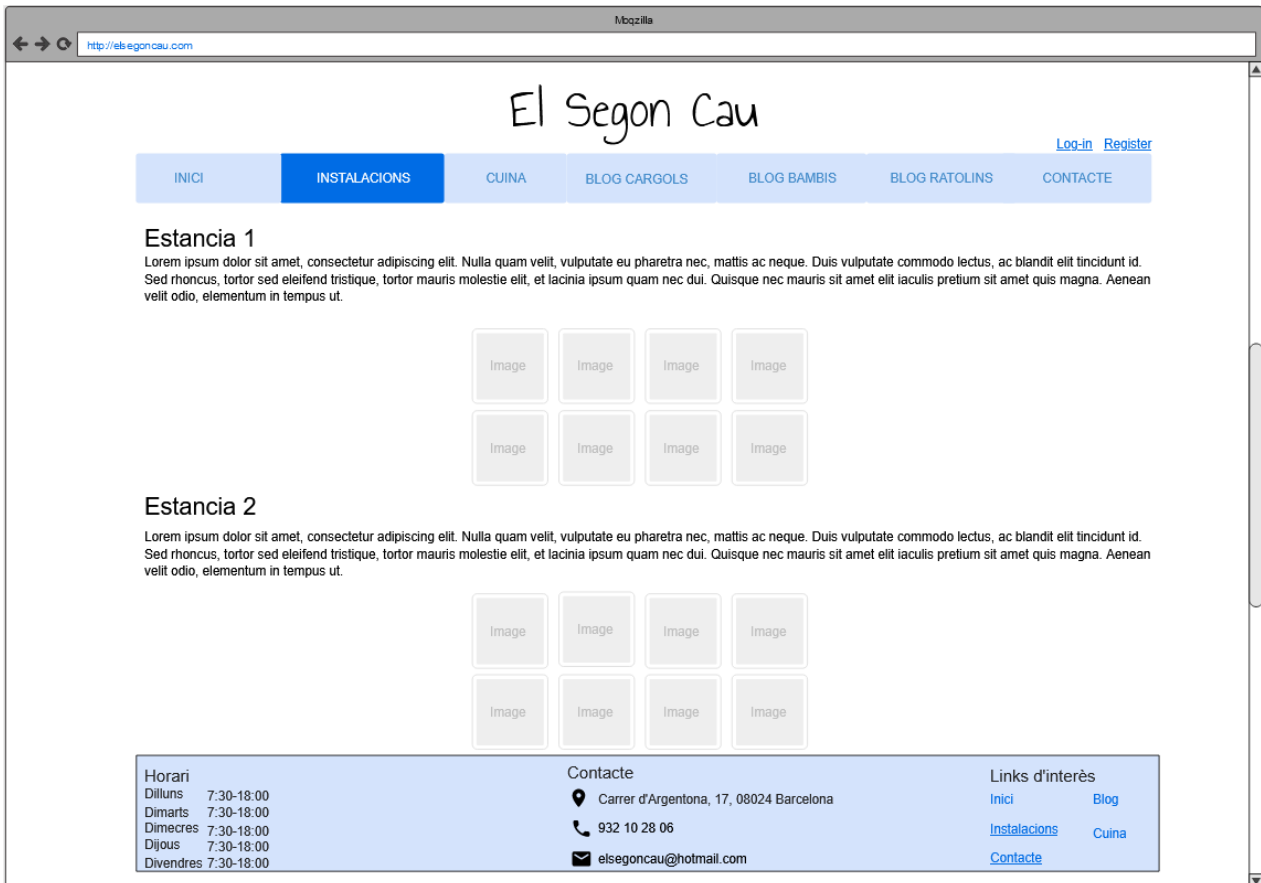
Ilustración 34: Mock up template Home, sección de información.



Ilustración 35: Mock up template Home, sección de avisos.

## Instalacions

El objetivo de esta template es mostrar las instalaciones del centro a los usuarios. Para cada estancia del centro educativo, se ofrecerá una breve descripción explicando sus características y un conjunto de imágenes.

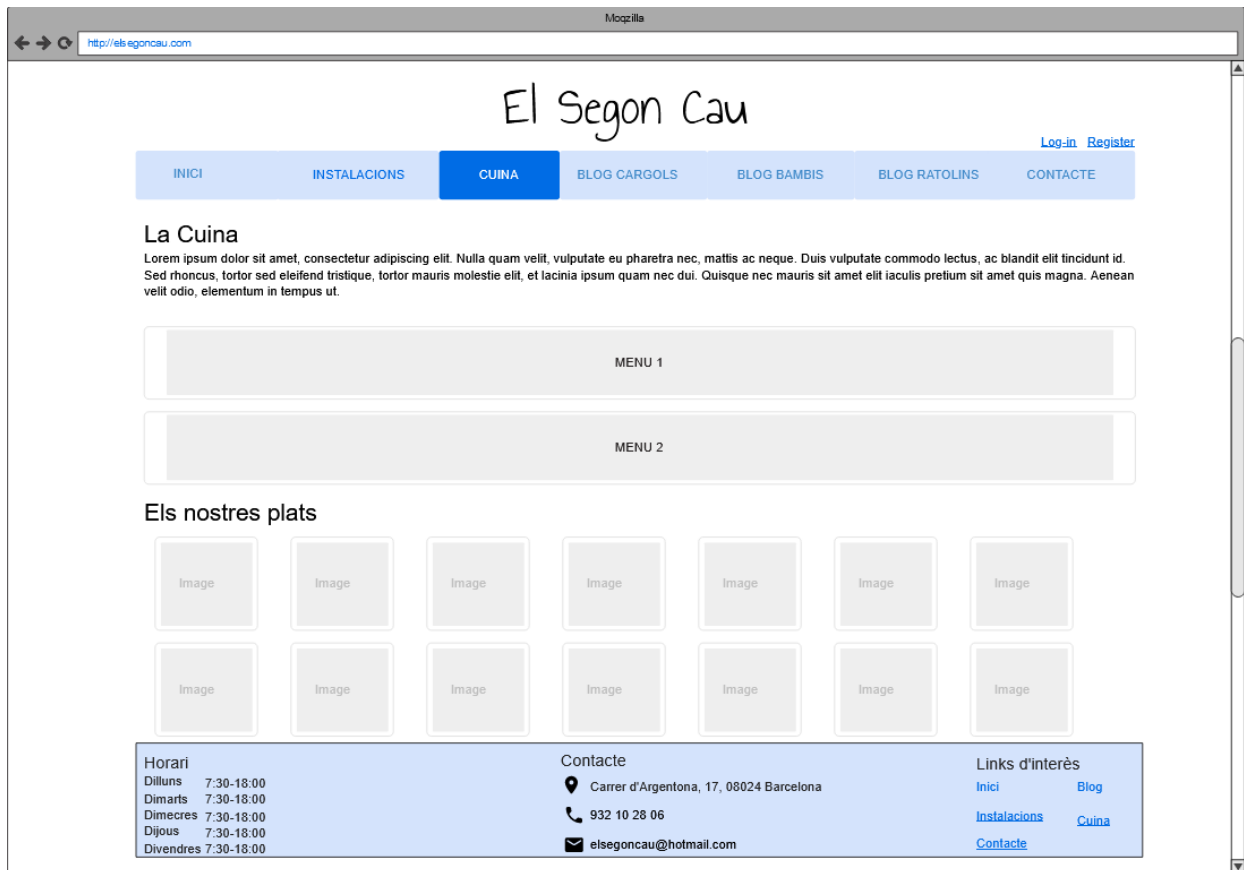


Il·lustració 36: Mock up template instalacions.



## Cuina

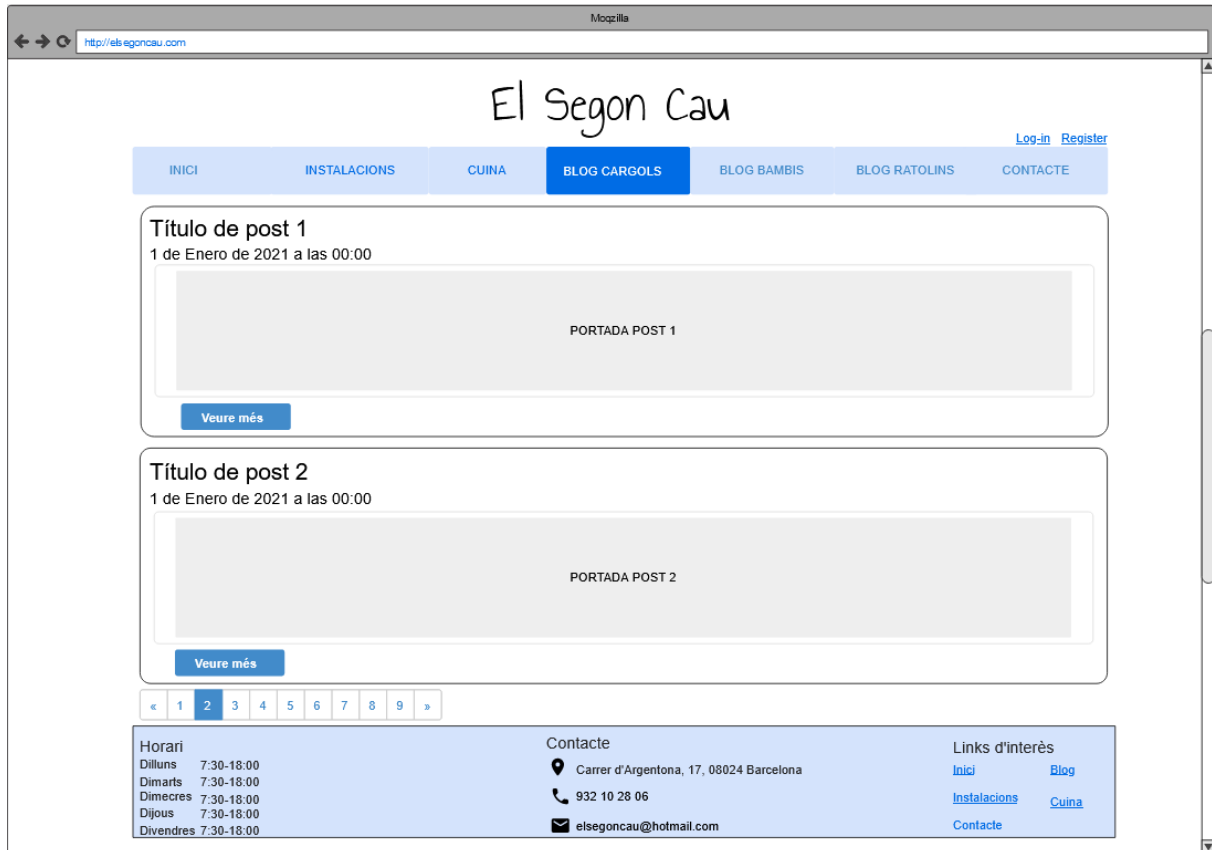
Aquí se ofrecerá información sobre la cocina y alimentación en el centro. Para que los padres sepan que comerán los alumnos, se mostrarán los diferentes menús que ofrece la guardería. También habrá un conjunto de fotografías con ejemplos de los platos que se cocinan en la escuela.



Il·lustració 37: Mock up template cuina.

## Blog

En la plantilla de los blogs se mostrarán todas las entradas relacionadas con una clase en concreto. El usuario verá el título de la publicación, la fecha de creación y una imagen relacionada. El usuario podrá ver el post de manera extensa si así lo desea pulsando un botón.



Il·lustración 38: Mock up template Blog.

## Blog Post

Esta template muestra el contenido de una entrada del blog en detalle. Se deberá mostrar el título, la fecha de su creación, una explicación si la tiene y sus imágenes asociadas. Al final del post estará la sección de comentarios donde se mostrarán en caso de haberlos.



Ilustración 39: Mock up template Blog Post

## Contacte

La página de contacto debe ofrecer la información de contacto del centro y los horarios de apertura. También contendrá un mapa con la localización del centro marcada. Por último encontraremos el formulario de contacto que el usuario podrá utilizar para comunicarse con la guardería.

El Segon Cau

[Log-in](#) [Register](#)

[INICI](#) [INSTALACIONS](#) [CUINA](#) [BLOG CARGOLS](#) [BLOG BAMBIS](#) [BLOG RATOLINS](#) [CONTACTE](#)

**Contacte**

📍 Carrer d'Argentona, 17, 08024 Barcelona

☎ 932 10 28 06

✉ elsegoncau@hotmail.com

**Horari**

Dilluns	7:30-18:00
Dimarts	7:30-18:00
Dimecres	7:30-18:00
Dijous	7:30-18:00
Divendres	7:30-18:00

Formulari de contacte

Nom

Email

Assumptes

Missatge

[Enviar](#)

**Horari**

Dilluns	7:30-18:00
Dimarts	7:30-18:00
Dimecres	7:30-18:00
Dijous	7:30-18:00
Divendres	7:30-18:00

**Contacte**

📍 Carrer d'Argentona, 17, 08024 Barcelona

☎ 932 10 28 06

✉ elsegoncau@hotmail.com

**Links d'interès**

[Inici](#) [Blog](#)

[Instalacions](#) [Cuina](#)

[Contacte](#)

Ilustración 40: Mock up template contacte

# 8. Validación

Para asegurar el buen funcionamiento de la aplicación y su calidad, se han ido haciendo validaciones durante el desarrollo e implementación de las funcionalidades.

## 8.1 Modo Debug

Cometer errores a la hora de programar es normal y más si tenemos en cuenta la poca experiencia previa con Django y con los web frameworks en general por parte del desarrollador. Por suerte, Django una herramienta para el trazado de errores que facilita el trabajo a la hora de trazarlos y corregirlos. Esta herramienta es el modo *debug*. Con el modo debug activado, cuando Django se tope con algún error o excepción que impide el correcto funcionamiento de nuestra aplicación, mostrará por pantalla una página con información detallada sobre ello. Información sobre el tipo de error, el nombre, la localización dentro del proyecto, los pasos que se han ejecutado hasta llegar a él, etc.



Ilustración 41: Pantalla de error del modo debug. En este caso se está intentando cargar una template inexistente.

Esta funcionalidad ha sido muy útil a lo largo del desarrollo del proyecto pero en especial en las etapas iniciales del desarrollo, donde no se tenía tanto conocimiento sobre la sintaxis específica de Django o sobre el orden de acciones correcto que se debía seguir.

## 8.2 Validación de funcionalidades

El modo debug, pese a ser de una gran utilidad, sólo tiene en cuenta errores a la hora de programar e implementar que imposibilitan la correcta ejecución de la aplicación. Esto no es suficiente, pues el código de una de las funcionalidades puede ejecutarse de manera correcta pero producir resultados inesperados o que no se corresponden con lo que necesitamos. Una vez acabada de implementar una de las funcionalidades, es necesario realizar pruebas funcionales mediante su uso para corroborar su correcto

funcionamiento y que cumplan con los diversos escenarios especificados en sus casos de uso.

Una de las herramientas que nos puede ayudar a gestionar este tipo de errores es el *issue tracker* integrado en Taiga. Un *issue tracker* es un sistema de seguimiento de incidentes. Nos permite crear incidencias o *issues* sobre un error o bug en nuestra aplicación, de esta manera dejaremos constancia de él para que pueda ser solucionado.

#40 El mapa con la ubicación del centro no se adapta bien a la pantalla cuando se cambia su tamaño.  
WEBAPP EL SEGON CAU ISSUE

+ Add tag

Created by Pablo Iniesta Romero  
07 Jan 2021 20:39

Al cambiar el tamaño de la ventana, el mapa no se reescala y se sale de su área.

Markdown Mode

0 attachments  
Drop attachments here!

0 Comments 1 Activities

Type a new comment here

CLOSED CLOSED

Bug type  
Normal severity  
Normal priority

ASSIGNED  
Pablo Iniesta Romero  
+ Add assigned

WATCHERS  
+ Add watchers Watch

VOTES 0

Ilustración 42: Ejemplo de una issue en Taiga.

Taiga nos permite asignarle un título, una explicación, el tipo de issue que en nuestro caso será bug, la severidad y la prioridad para solucionarlo. A pesar de que esta herramienta está pensada para ser usadas por equipos de varias personas, sigue siendo útil como elemento organizativo en un proyecto individual.

## 9. Planificación temporal

En todo proyecto de software, una buena planificación temporal es de vital importancia para garantizar el buen desarrollo de este e incrementar sus probabilidades de éxito y aún más si existe una fecha límite de entrega fija como es el caso.

En este punto se mostrará la planificación temporal que se ideó al inicio del proyecto. También veremos los posibles cambios y adaptaciones que han surgido en el transcurso del desarrollo de la aplicación

### 9.1 Planificación temporal inicial

La duración del proyecto es de 4 meses aproximadamente, empezando el día 21 de Septiembre de 2020 y terminando el día 26 de Enero de 2021, que es cuando se realizará la defensa de este proyecto. Esto nos deja con un periodo de 126 días aproximadamente y con una duración aproximada de 500 horas en total.

El proyecto se dividirá en diferentes etapas, dependiendo del tipo de trabajo a realizar en cada una de ellas. La etapa inicial se corresponde con la gestión del proyecto, donde se analizan aspectos como el alcance del proyecto, la planificación temporal, los presupuestos... Al seguir una metodología ágil para desarrollar el proyecto, la etapa de desarrollo e implementación será dividida en sub-etapas o sprints, en concreto 4 sprints donde se tratarán los diferentes aspectos de la aplicación web. Para cada sprint, aparte de la implementación de las funcionalidades de la web, también se llevará a cabo la redacción de la documentación correspondiente a lo que se haya implementado y al final de cada sprint se hará una sesión retrospectiva para evaluar el rendimiento e identificar posibles puntos a mejorar o a continuar en las siguientes etapas. De ser necesario, se tendrá una reunión con el director del TFG al final de cada sprint para ir informando sobre la evolución del trabajo. Por último, se dedicará otra etapa a la preparación de la memoria del trabajo así como a preparar la defensa.

#### 9.1.1 Fases del proyecto y definición de tareas

A continuación se definirán las tareas a realizar en cada una de las etapas del proyecto. Podemos diferenciar entre 4 tipos de tareas: Tareas de gestión de proyecto (GP), tareas de desarrollo (DES), tareas de documentación (DOC) y tareas de validación y despliegue (VD).

#### **Gestión del proyecto**

Esta etapa se corresponde con el curso de gestión de proyectos que se realiza al principio del cuatrimestre. Su objetivo principal es idear, definir y empezar a documentar el proyecto de final de grado.

**GP1 –Contexto y alcance:** Contextualizar la propuesta del trabajo de final de grado y acotar su alcance para así definir cuanto abarcará el proyecto.

**GP2–Planificación temporal:** Definir las diferentes tareas a realizar a lo largo del proyecto y su estimación temporal. También se plantean posibles riesgos y cómo podrían afrontarse.

**GP3–Presupuesto y sostenibilidad:** Analizar los posibles costes que podría tener llegar a realizar el proyecto en caso de ser una empresa. También se analiza la sostenibilidad del proyecto.

**GP4–Documentación final:** Incluir los 3 documentos anteriores en un mismo documento cambiando, adaptando y corrigiendo todo aquello que sea necesario en función al feedback obtenido previamente.

## **Sprint 1**

En el primer sprint se preparará el entorno de trabajo y todo lo necesario para empezar a implementar la aplicación. Se implementará el sistema de usuarios y una primera estructura de la web.

**DES1–Preparación del entorno de trabajo:** Preparar el entorno de trabajo: descargar los programas necesarios, los lenguajes, las librerías, crear proyecto de GitHub y Taiga.

**DES2–Sistema de usuarios:** Desarrollar el sistema de usuarios. Permitir el registro, login y log-out de usuarios, crear diferentes tipos de usuario (usuario regular y usuario profesor), permitir al usuario actualizar su información.

**DES3–Estructura aplicación web:** Identificar todos los apartados y sub-apartados que serán necesarios para la aplicación web e implementarlos permitiendo la navegación entre pantallas para así contar con una base o esqueleto de la aplicación donde poder trabajar.

**GP5–Retrospección Sprint 1:** Analizar el trabajo realizado durante el sprint e identificar puntos a mejorar o puntos a mantener para los siguientes sprints. Realizar cambios en la metodología o planificación del proyecto si es necesario.

**GP6–Reuniones sprint 1:** De ser necesario, se realizarán reuniones con el director de este proyecto para informar sobre su progreso y resolver posibles dudas o preguntas.

**DOC1–Documentación sprint 1:** Redactar la documentación referente al sprint 1.

## **Sprint 2**

Este sprint se centrará en implementar la funcionalidad de blog, por tanto se trabajará en los blog posts y en los comentarios.

**DES4–Sistema de posts:** Implementar la creación, edición, eliminación y visualización de posts en las secciones de las clases de la guardería. Podrán contener imágenes, videos y texto.



**DES5–Sistema de comentarios:** Implementar la creación, eliminación y visualización de comentarios en los posts. Podrán contener texto. Permitir la moderación de comentarios a los usuarios administradores.

**GP7–Retrospección Sprint 2:** Analizar el trabajo realizado durante el sprint 2 e identificar puntos a mejorar o puntos a mantener para los siguientes sprints. Realizar cambios en la metodología o planificación del proyecto si es necesario.

**GP8–Reuniones sprint 2:** De ser necesario, se realizarán reuniones con el director de este proyecto para informar sobre su progreso y resolver posibles dudas o preguntas.

**DOC2–Documentación sprint 2:** Redactar la documentación referente al sprint 2.

### **Sprint 3**

El tercer sprint se centrará en el aspecto informativo de la aplicación web, es decir, la información y formulario de contacto y la página principal.

**DES6–Formulario de contacto:** Implementar un formulario de contacto para facilitar la comunicación directa con la guardería. Se podrá utilizar tanto si se está registrado como si no.

**DES7–Implementar localización con Google Maps:** Implementar el uso de Google Maps en nuestra aplicación para ver la localización del centro usando el api de Google.

**DES8–Página principal o Home:** Implementar la página principal con toda la información del centro necesaria y unirla a todas las funcionalidades ya existentes.

**GP9–Retrospección Sprint 3:** Analizar el trabajo realizado durante el sprint 3 e identificar puntos a mejorar o puntos a mantener para los siguientes sprints. Realizar cambios en la metodología o planificación del proyecto si es necesario.

**GP10–Reuniones sprint 3:** De ser necesario, se realizarán reuniones con el director de este proyecto para informar sobre su progreso y resolver posibles dudas o preguntas.

**DOC3–Documentación sprint 3:** Redactar la documentación referente al sprint 3.

### **Sprint 4**

El último sprint está pensado para mejorar la interfaz gráfica de la aplicación web y validar la aplicación.

**DES9–Mejora de interfaz gráfica y usabilidad:** Mejorar la usabilidad de todas las pantallas de la aplicación web y sus interfaces gráficas.

**VD1–Validación de la aplicación:** Comprobar que el proyecto cumple todos los objetivos y requisitos definidos previamente y que cumple con las expectativas del centro.

**VD2–Despliegue:** Mostrar todas las funcionalidades de la aplicación a los empleados del centro y enseñarles cómo hacer uso de ellas.

**GP11–Retrospección Sprint 4:** Analizar el trabajo realizado durante el sprint 4 e identificar puntos a mejorar o puntos a mantener. Realizar cambios en la metodología o planificación del proyecto si es necesario.

**GP12–Reuniones sprint 4:** De ser necesario, se realizarán reuniones con el director de este proyecto para informar sobre su progreso y resolver posibles dudas o preguntas.

**DOC4–Documentación sprint 4:** Redactar la documentación referente al sprint 4.

## Documentación final

La última etapa está destinada a acabar la documentación del proyecto y a preparar la defensa.

**DOC5–Documentación definitiva memoria:** Acabar de redactar la memoria del proyecto para dejarla lista para entregar

**DOC6–Preparar defensa proyecto:** Preparar y practicar la defensa del proyecto para exponerlo ante tribunal

<i>Código</i>	<i>Nombre</i>	<i>Tiempo</i>	<i>Dependencias</i>
<i>GP1</i>	Contexto y alcance	25	-
<i>GP2</i>	Planificación temporal	15	GP1
<i>GP3</i>	Presupuesto y sostenibilidad	20	GP2
<i>GP4</i>	Documentación final	15	GP3
<i>DES1</i>	Preparar entorno trabajo	10	-
<i>DES2</i>	Sistema de usuarios	40	DES1
<i>DES3</i>	Estructura aplicación web	30	DES2
<i>GP5</i>	Retrospección Sprint 1	5	DES3
<i>GP6</i>	Reuniones Sprint 1	4	DES3
<i>DOC1</i>	Documentación Sprint 1	15	-
<i>DES4</i>	Sistema de posts	40	Sprint 1
<i>DES5</i>	Sistema de comentarios	30	DES4
<i>GP7</i>	Retrospección Sprint 2	5	DES5
<i>GP8</i>	Reuniones Sprint 2	4	DES5
<i>DOC2</i>	Documentación Sprint 2	15	-
<i>DES6</i>	Formulario de contacto	20	Sprint2
<i>DES7</i>	Localización Google Maps	25	DES6
<i>DES8</i>	Página principal	25	DES7
<i>GP9</i>	Retrospección Sprint 3	5	DES8
<i>GP10</i>	Reuniones Sprint 3	4	DES8
<i>DOC3</i>	Documentación Sprint 3	15	-
<i>DES9</i>	Interfaz gráfica y usabilidad	60	Sprint 3
<i>VD1</i>	Validación de la aplicación	10	DES9
<i>VD2</i>	Despliegue	5	DES9, VD1
<i>GP11</i>	Retrospección Sprint 4	5	DES9
<i>GP12</i>	Reuniones Sprint 4	4	DES9
<i>DOC4</i>	Documentación Sprint 4	15	-
<i>DOC5</i>	Documentación Memoria	25	Sprint 4
<i>DOC6</i>	Preparar defensa	5	DOC5

## 9.1.2 Gantt inicial

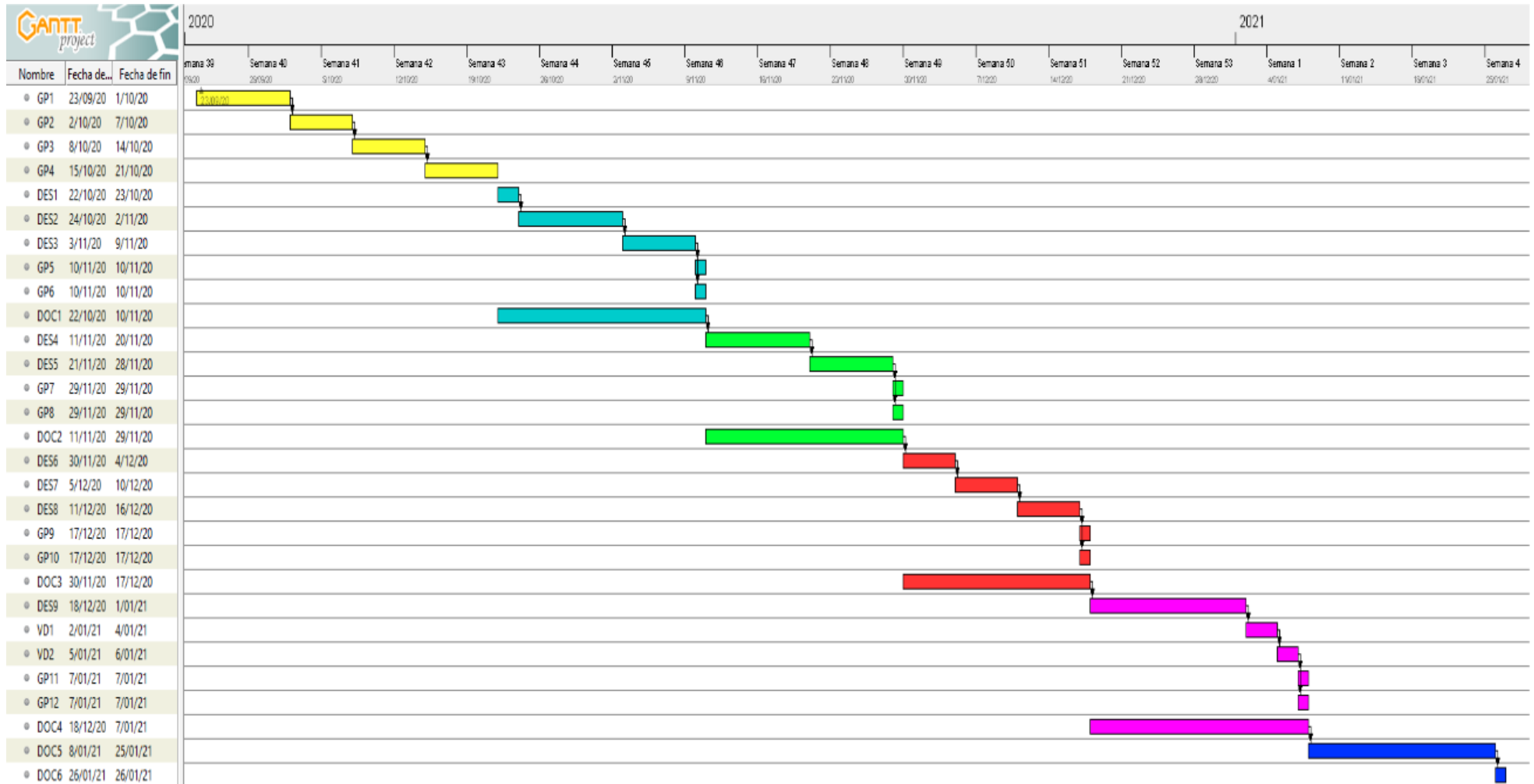


Ilustración 43: Diagrama Gantt inicial

## 9.1.2 Gestión del riesgo

Al hacer uso de tecnologías con las que no se es familiar y al desarrollar una aplicación como esta por primera vez de manera individual, es muy posible que a lo largo del desarrollo se encuentren obstáculos o factores de riesgo que pueden llegar a cambiar el desarrollo del trabajo. Es necesario prever estos obstáculos para minimizar en la mayor medida posible su impacto.

### **Inexperiencia en las tecnologías a utilizar**

El hecho de utilizar tecnologías que no son familiares para el desarrollador puede suponer que algunas tareas se demoren más de lo normal pues se tendrá que aprender desde cero, por ello la duración de las tareas de desarrollo ha sido sobreestimada para que esta posible demora no afecte al desarrollo total.

### **Fallo del equipo de trabajo**

Todos los programas y herramientas necesarias para la realización del trabajo se instalarán en un ordenador secundario. En caso de fallo en el equipo principal, la demora en el trabajo será mínima pues el equipo secundario estará listo.

Este riesgo en concreto se hizo realidad en 2 ocasiones a lo largo del proyecto. Un error con una de las actualizaciones de Windows hizo que fuese necesario formatear el ordenador principal de trabajo. En una segunda ocasión el disco duro del ordenador dejó de funcionar, siendo necesario su reemplazo por uno nuevo. Por suerte estos inconvenientes no supusieron ningún retraso en el desarrollo del proyecto pues el ordenador secundario estaba listo para su uso mientras el principal era reparado.

### **Dificultades imprevistas**

Si se llega a dar algún imprevisto no contemplado previamente, se podría llegar a reducir la duración de la última fase (Documentación de la memoria) para acomodar en la mayor medida posible el retraso que estos imprevistos puedan causar, llegando a la fecha de entrega sin problema.

## 9.2 Cambios respecto a la planificación inicial

El desarrollo del proyecto ha respetado en gran medida la planificación inicial. Se han seguido los sprints y se han implementado las diversas funcionalidades en el orden especificado. Aun así, a lo largo del desarrollo del proyecto se han producido cambios respecto a la planificación inicial. Por petición del centro, se incluyeron dos funcionalidades nuevas que en un estado inicial no se tuvieron en cuenta, la sección de instalaciones con galería de imágenes y la sección de cocina con los menús y galería de imágenes de platos.

Este cambio queda reflejado en el sprint número 4. En vez de contar con una sola gran tarea de desarrollo centrada en la mejora de la interfaz gráfica, ahora contamos con 2 tareas previas más.

**DES9–Sección de instalaciones:** Implementar la sección de las instalaciones del centro con descripciones y galería de imágenes.

**DES10–Sección de cocina:** Implementar la sección de cocina del centro con información sobre los menús y galería de platos.

**DES11–Mejora de interfaz gráfica y usabilidad:** Mejorar la usabilidad de todas las pantallas de la aplicación web y sus interfaces gráficas.

Se decidió implementar estas dos nuevas tareas en el último sprint por un motivo. Durante el desarrollo de cada una de las funcionalidades del proyecto también se fue trabajando en la interfaz gráfica y sus diferentes elementos funcionales. Por ello, una sola tarea de tanta duración centrada en la interfaz gráfica no era necesaria. En la nueva planificación las 3 nuevas tareas se realizaron en el periodo que ocupaba la tarea de interfaz gráfica original, de esta manera este cambio no supuso ninguna alteración en las fechas planeadas del proyecto.

<i>Código</i>	<b>Nombre</b>	<b>Tiempo</b>	<b>Dependencias</b>
<i>GP1</i>	Contexto y alcance	25	-
<i>GP2</i>	Planificación temporal	15	GP1
<i>GP3</i>	Presupuesto y sostenibilidad	20	GP2
<i>GP4</i>	Documentación final	15	GP3
<i>DES1</i>	Preparar entorno trabajo	10	-
<i>DES2</i>	Sistema de usuarios	40	DES1
<i>DES3</i>	Estructura aplicación web	30	DES2
<i>GP5</i>	Retrospección Sprint 1	5	DES3
<i>GP6</i>	Reuniones Sprint 1	4	DES3
<i>DOC1</i>	Documentación Sprint 1	15	-
<i>DES4</i>	Sistema de posts	40	Sprint 1
<i>DES5</i>	Sistema de comentarios	30	DES4
<i>GP7</i>	Retrospección Sprint 2	5	DES5
<i>GP8</i>	Reuniones Sprint 2	4	DES5
<i>DOC2</i>	Documentación Sprint 2	15	-
<i>DES6</i>	Formulario de contacto	20	Sprint2
<i>DES7</i>	Localización Google Maps	25	DES6
<i>DES8</i>	Página principal	25	DES7
<i>GP9</i>	Retrospección Sprint 3	5	DES8
<i>GP10</i>	Reuniones Sprint 3	4	DES8
<i>DOC3</i>	Documentación Sprint 3	15	-
<i>DES9</i>	Sección de instalaciones	20	Sprint 3
<i>DES10</i>	Sección de cocina	20	DES9
<i>DES11</i>	Interfaz gráfica y usabilidad	20	DES10
<i>VD1</i>	Validación de la aplicación	10	DES9
<i>VD2</i>	Despliegue	5	DES9, VD1
<i>GP11</i>	Retrospección Sprint 4	5	DES9
<i>GP12</i>	Reuniones Sprint 4	4	DES9
<i>DOC4</i>	Documentación Sprint 4	15	-
<i>DOC5</i>	Documentación Memoria	25	Sprint 4
<i>DOC6</i>	Preparar defensa	5	DOC5

Tabla 2: Tabla de tareas del proyecto de la planificación final

## 9.2.1 Gantt final

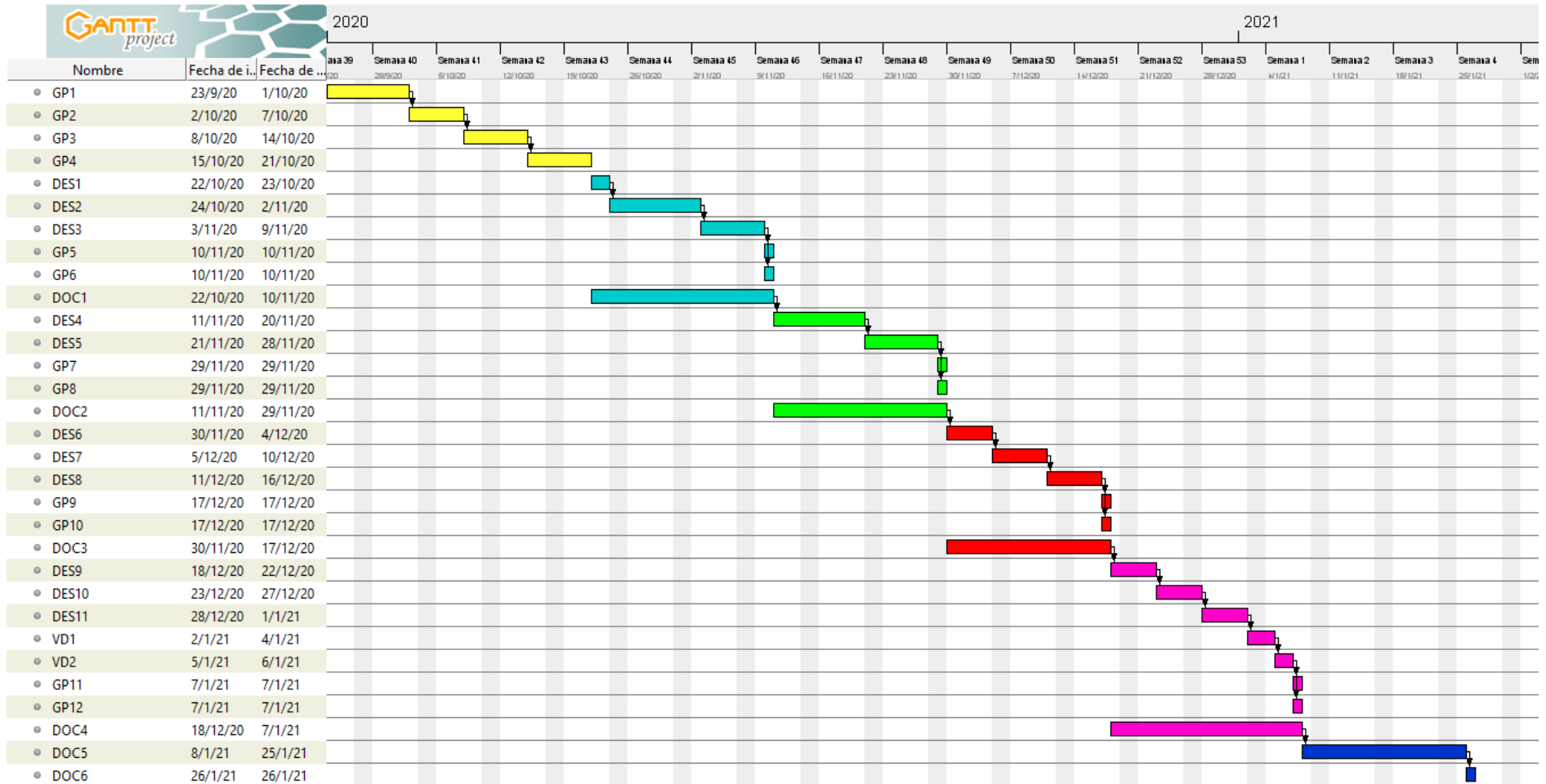


Ilustración 44: Diagrama Gantt final

## 10. Estimación de costes

Tras realizar la planificación temporal, podemos continuar con la gestión económica del proyecto. La aplicación web ha sido desarrollada en su plenitud por un solo desarrollador y de manera individual, sin ninguna empresa de desarrollo de software de por medio. Si hiciésemos la estimación de costes de la situación real, no habría apenas elementos a tener en cuenta. Es por ello que en este punto se realizará una estimación de costes del proyecto partiendo de una empresa ficticia, con todo lo que ello conlleva.

### 10.1 Recursos humanos

La fuerza de trabajo humana es quizá el aspecto más importante de todo proyecto software pues sin ella su correcto desarrollo no sería posible. Para calcular de manera correcta la cantidad necesaria para cubrir los gastos relacionados con el personal, primero es necesario averiguar cuantas horas invertirá cada rol en el proyecto. Contamos con 4 roles diferentes: Jefe de proyecto, ingeniero de requisitos, programador y tester. Las tareas de desarrollo están llevadas a cabo por los tres últimos y por tanto es necesario especificar el porcentaje de horas que invertirá cada uno en cada una de ellas. Se establece que para cada tarea de desarrollo el ingeniero de requisitos realizará el 15% de las horas, el programador el 65% y el tester el 20% restante.

Para calcular el coste total de cada una de las tareas se ha obtenido el sueldo medio por hora de cada uno de los roles del proyecto [13]:

- Jefe de proyecto: 21.5€ brutos, 27.95€ netos.
- Ingeniero de requisitos: 18€ brutos, 23.4€ netos.
- Programador: 15€ brutos, 19.5€ netos.
- Tester: 13€ brutos, 17€ netos.

Una vez tenemos los sueldos de los trabajadores y el porcentaje de trabajo que realizará cada uno de ellos, podemos calcular para cada tarea especificada en el apartado de planificación temporal el número de horas y su coste por rol y en total.

<i>Código</i>	<b>Jefe de proyecto</b>	<b>Ingeniero de requisitos</b>	<b>Programador</b>	<b>Tester</b>	<b>Total</b>
<i>GP1</i>	25h / 698.75€				25h / 698.75€
<i>GP2</i>	15h / 419.25€				15h / 419.25€
<i>GP3</i>	20h / 698.75€				20h / 698.75€
<i>GP4</i>	15h / 419.25€				15h / 419.25€
<i>DES1</i>		1.5h / 35.1€	6.5h / 126.75€	2h / 34€	10h / 195.85€
<i>DES2</i>		6h / 140.4€	26h / 507€	8h / 136€	40h / 783.4€
<i>DES3</i>		4.5h / 105.3€	19.5h / 380.25€	6h / 102€	30h / 587.55€
<i>GP5</i>	5h / 139.75€				5h / 139.75€
<i>GP6</i>	4h / 111.8€				4h / 111.8€
<i>DOC1</i>	15h / 419.25€				15h / 419.25€
<i>DES4</i>		6h / 140.4€	26h / 507€	8h / 136€	40h / 783.4€
<i>DES5</i>		4.5h / 105.3€	19.5h / 380.25€	6h / 102€	30h / 587.55€

GP7	5h / 139.75€				5h / 139.75€
GP8	4h / 111.8€				4h / 111.8€
DOC2	15h / 419.25€				15h / 419.25€
DES6		3h / 70.2€	13h / 253.5€	4h / 68€	20h / 391.7€
DES7		3.75h / 87.75€	16.25h / 316.8€	5h / 85€	25h / 489.55€
DES8		3.75h / 87.75€	16.25h / 316.8€	5h / 85€	25h / 489.55€
GP9	25h / 489.55€				5h / 139.75€
GP10	4h / 111.8€				4h / 111.8€
DOC3	15h / 419.25€				15h / 419.25€
DES9		3h / 70.2€	13h / 253.5€	4h / 68€	20h / 391.7€
DES10		3h / 70.2€	13h / 253.5€	4h / 68€	20h / 391.7€
DES11		3h / 70.2€	13h / 253.5€	4h / 68€	20h / 391.7€
VD1		1.5h / 35.1€	6.5h / 126.75€	2h / 34€	10h / 195.85€
VD2	5h / 139.75€				5h / 139.75€
GP11	5h / 139.75€				5h / 139.75€
GP12	4h / 111.8€				4h / 111.8€
DOC4	15h / 419.25€				15h / 419.25€
DOC5	25h / 698.75€				25h / 698.75€
DOC6	5h / 139.75€				5h / 139.75€
TOTAL	206h / 5757.7€	43.5h / 1017.9€	188.5h / 3675.75€	58h / 986€	496h / 11437.35€

Tabla 3: Horas y sueldos netos por rol y tarea a realizar

## 10.2 Recursos materiales

También es necesario calcular el coste de los recursos materiales necesarios para realizar el proyecto.

**PyCharm:** 199€ por año y por usuario, por tanto se necesitarán 2 licencias para el programador y el tester. 398€ en total. [14].

**Portátil:** Se utilizará un portátil Lenovo valorado en 800€ para el desarrollo del proyecto. Es necesario calcular su amortización en el transcurso del proyecto mediante la siguiente fórmula:

$$\frac{\text{Coste Portátil}}{\text{Vida útil} * \text{días hábiles} * \text{horas al día}} * \text{Horas Proyecto} = \frac{800€}{4 \text{ años} * 220 \text{ días} * 4} * 496 = 112.7€$$

Consideramos que la vida útil de un portátil es de 4 años, los días hábiles en un año son 220 y se utilizará durante 4 horas al día durante 496 horas. Obtenemos una amortización de 112.7€. Tanto el programador como el tester necesitaran un equipo para trabajar, por tanto debemos multiplicar esta amortización por 2, en total 225.4€.

## 10.3 Recursos generales

A parte de los recursos humanos y materiales, también es necesario tener en cuenta el coste de las instalaciones donde se llevaría a cabo el desarrollo del proyecto. Como se ha supuesto que la aplicación se llevará a cabo por 4 personas o roles diferentes, es necesario buscar una oficina acorde a sus necesidades. Una pequeña oficina con acceso



en cualquier momento del día, los 7 días de la semana y con internet ilimitado ronda los 302.5€ al mes [15].

## 10.4 Contingencias

Al tratarse de un proyecto bastante estándar, se fijara una contingencia del 10% para hacer frente a posibles gastos imprevistos que puedan surgir a lo largo del desarrollo de la aplicación.

<b>Tipo</b>	<b>Coste</b>	<b>Coste Final</b>
Recursos humanos	11437.35€	12581.1€
Hardware	225.4€.	247.94€
Software	398€	437.8€
Recursos generales	302.5€	332.75€

Tabla 4: Contingencias del proyecto

## 10.5 Coste total

A continuación podemos ver el presupuesto final del desarrollo del proyecto.

<b>Tipo de coste</b>	<b>Coste final</b>
Recursos humanos	12581.1€
Recursos materiales	685.74€
Recursos generales	332.75€
Coste total	13599.59€

Tabla 5: Coste total del proyecto.

# 11. Sostenibilidad

Durante el transcurso de la carrera y en las diferentes asignaturas que se van cursando se menciona y se estudia la importancia de los diferentes aspectos que hacen o no que un proyecto sea sostenible. Aspectos como el impacto social, medioambiental o económico que supone realizar un proyecto en concreto. Es cierto que en su momento el alumnado, en el cual me incluyo, puede no darle la importancia que se merece a la sostenibilidad pero es a la hora de realizar proyectos como este TFG cuando uno se da cuenta de lo esencial que es tenerla en cuenta.

Por lo que respecta al aspecto económico, se aprende a analizar los posibles gastos de un proyecto y calcular el coste total de su desarrollo y mantenimiento gracias a asignaturas como GPS y PES que hacen gran hincapié en la necesidad de analizar la viabilidad económica de estos.

Otro aspecto importante a tener en cuenta a la hora de desarrollar un proyecto es el social. Analizar los pros y los contras de la realización de un proyecto de software desde un punto de vista ético, su impacto en la sociedad, el grado de accesibilidad... siempre con la mayor transparencia posible y respetando la legislación vigente.

Por último tenemos el aspecto medioambiental que es posiblemente el más olvidado de todos ellos. A lo largo de la carrera se han ofrecido varias charlas, actividades o visionado de documentales sobre el impacto de la industria informática en el medio ambiente pero sobre todo se hace referencia a la fabricación de hardware.

## 11.1 Dimensión económica

Tras realizar los cálculos pertinentes, el presupuesto final del proyecto se estima en unos 13599.59€. Esta estimación está hecha desde el punto de vista de una empresa que se dedica a realizar proyectos de software de manera profesional con diferentes roles y personal para el proyecto y sueldos competitivos. Este coste no supondría un problema si la empresa interesada en la creación de la aplicación web facturase una gran cantidad de dinero pero en el contexto de este proyecto, un gasto de 13599.59€ no sería viable para una guardería como El Segon Cau.

Por tanto lo ideal sería que el proyecto fuese realizado por algún tipo de desarrollador independiente o por alguna empresa de desarrollo de aplicaciones web más pequeña reduciendo así la factura final del proyecto. En nuestro caso, al tratarse de un TFG y al ser El Segon Cau un colegio muy cercano al autor del trabajo, no se realizará ningún cobro ni será necesario ningún aporte monetario por la implementación de la aplicación web.

Otro punto a tener en cuenta será la disponibilidad del código fuente de la aplicación de manera gratuita mediante el repositorio de github, haciendo posible modificar o añadir nuevas funcionalidades en un futuro de manera sencilla y con coste mínimo, pues no se necesitará empezar desde cero.

## 11.2 Dimensión medioambiental

Al tratarse de un proyecto de desarrollo de software bastante estándar, el impacto medioambiental que pueda producir no será muy grande. Podríamos contar como impacto negativo los recursos materiales necesarios para desarrollarlo como la electricidad o la creación de los equipos de trabajo necesarios.

Por otro lado, la realización del proyecto puede llegar a impactar de manera positiva el medioambiente. Por ejemplo, la guardería utiliza papel a la hora de informar a los padres sobre los posibles eventos o noticias mediante circulares. Con la aplicación web este gasto de papel se podrá ahorrar informando a los padres online.

## 11.3 Dimensión social

Con la realización de este proyecto, aparte de profundizar en los conocimientos de desarrollo de proyectos de software y, en concreto, en el desarrollo de páginas y aplicaciones web mediante uso de frameworks desconocidos hasta ahora, también se ha aprendido a tratar y a relacionarse de manera profesional con las diferentes partes implicadas del proyecto como el director del TFG o la escuela El Segon Cau.

El hecho de implementar la aplicación web para El Segon Cau, aparte de aumentar los clientes potenciales del centro y mejorar su imagen, mejorará la comunicación padres-escuela. Las familias podrán seguir con más facilidad el día a día de sus hijos y conocer de manera más cercana y familiar a la escuela y a los empleados, aumentando su tranquilidad y calidad de vida pues podrán monitorizar una de las etapas más importantes para el desarrollo de sus infantes. Por tanto, la realización de este proyecto es necesaria ya que dotara a El Segon Cau de una herramienta que mejorará la situación actual de todas las partes implicadas, tanto de manera económica como social.

## 12. Regulaciones y legislación

Al girar este trabajo en torno a la aplicación web de una guardería que además contará con un apartado de un Blog donde se documentarán las actividades que realizan los alumnos, el aspecto más sensible y quizás el más importante a tener en cuenta es la ley de protección de datos y los derechos de imagen.

El *Reglamento General de Protección de Datos* (GDPR) que se aplica en el estado español mediante la *Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales* [16] dictamina que, en lo que respecta al tratamiento de datos de carácter personal se debe, independientemente de la nacionalidad o residencia del individuo, respetar sus libertades y derechos fundamentales, en particular el derecho a la protección de los datos de carácter personal. En la aplicación web del Segon Cau, sólo el personal autorizado tendrá acceso a los datos que se pueden considerar de carácter personal y en ningún momento se hará pública dicha información ni se dará acceso a ella a empresas externas.

Por otro lado, el artículo 18 de la Constitución Española garantiza el derecho a la propia imagen como un derecho fundamental, desarrollándose su contenido en la *Ley Orgánica de protección civil del derecho al honor, a la intimidad personal y familiar y a la propia imagen*. [17] Esta ley implica que se necesita el consentimiento explícito de una persona si se la quiere fotografiar y subir las imágenes a internet. Hace unos años El Segon Cau decidió dejó de pedir el consentimiento de los derechos de imagen de los alumnos. Esto significa que en ningún momento aparecerá en la aplicación web alguna imagen con la cara de un alumno o con cualquier rasgo característico que haga posible su identificación.

## 13. Conclusiones

A pesar de las posibles dificultades e imprevistos que han ido surgiendo a lo largo del desarrollo del proyecto, se han cumplido todos los objetivos propuestos en la etapa inicial, llegando a incluir nuevas funcionalidades que no se tenían en cuenta al principio del desarrollo.

En el momento de la defensa de este proyecto y, por tanto, la finalización de este trabajo de final de grado, la aplicación queda diseñada y todas las funcionalidades quedan implementadas. En este estado, la aplicación queda lista para la siguiente etapa, ser desplegada en internet.

Durante el desarrollo del proyecto se ha mantenido una muy buena relación con El Segon Cau y su equipo de dirección. En el transcurso de estos 4 meses se han ido organizando pequeñas reuniones para informar sobre el estado de la aplicación intentando tener en cuenta al máximo las peticiones y los gustos de las representantes del centro. Especialmente en la etapa de desarrollo de la interfaz gráfica, donde la colaboración centro-desarrollador fue máxima. También se organizó un encuentro con las empleadas del centro para explicarles el funcionamiento del nuevo sistema y de su papel a la hora de mantener y actualizar el contenido. Las profesoras recibieron los cambios de manera muy positiva remarcando sobre todo la sencillez de las tareas que debían realizar y el aspecto colorido de la web. De este modo podemos afirmar que al acabar este proyecto ambas partes implicadas, tanto el desarrollador como la empresa El Segon Cau, están satisfechos con los resultados obtenidos.

Es cierto que el desarrollo del proyecto no fue del todo simple, sobre todo al empezar la implementación, pues casi todas las tecnologías y herramientas a utilizar eran nuevas para mí. Una vez asentadas las bases y el conocimiento sobre Django, HTML y CSS el desarrollo comenzó a ir mucho más fluido permitiendo seguir la planificación del trabajo sin atrasos.

Gracias a este proyecto he podido profundizar en el inmenso mundo del desarrollo de aplicaciones web tras quedarme con ganas de más al realizar la asignatura de la especialidad de Software ASW. A parte de la satisfacción y la realización personal que implica completar un proyecto como este, espero poder aplicar todo este nuevo conocimiento y habilidades adquiridas en este periodo de tiempo una vez pase a formar parte del excitante mundo laboral asociado al desarrollo de software al terminar la carrera.

También estoy agradecido por poder realizar un trabajo planteado por mí mismo y por dotar de una cierta libertad a la hora de trabajar y tomar decisiones sobre él. Es todo un placer poder realizar una aplicación que mejorará la situación actual de El Segon Cau, un centro muy cercano a mí y al que tengo mucho aprecio desde pequeño.

## 13.1 Futuro

La finalización de este trabajo de final de grado no implica el fin de la evolución de la aplicación web. En el mes posterior a la defensa del proyecto se seguirá trabajando con El Segon Cau para buscar una empresa de hospedaje de aplicaciones web que se adapte a la perfección a las necesidades de la aplicación. Además y tal y como se ha visto previamente, el producto está pensado para ser utilizado a diario por parte de los empleados del centro y de los usuarios, aumentándose y renovándose el contenido de la web de forma periódica y sin la necesidad de la participación del desarrollador.

Al tener acceso al código fuente y gracias a que la instalación del espacio de trabajo es relativamente sencilla, no debería suponer ningún tipo de problema ir añadiendo funcionalidades nuevas con el transcurso del tiempo y a petición del centro. Sobre todo una vez este desafortunado periodo marcado por el coronavirus que nos afecta a todos por igual llegue a su fin y todo pueda volver a funcionar con normalidad.

## 13.2 Competencias técnicas

A continuación se citarán y justificarán las diferentes competencias técnicas del proyecto.

### **CES1.4: Desarrollar, mantener y evaluar servicios y aplicaciones distribuidas con soporte de red.**

El proyecto gira en torno al desarrollo de una aplicación web. Toda la tecnología utilizada está relacionada con el soporte de red en mente y está conectada entre sí. Desde el frontend con HTML, CSS y JS que interactúa con los navegadores y el usuario hasta el backend con Django que gestiona los procesos internos de la aplicación.

### **CES1.5: Especificar, diseñar, implementar y evaluar bases de datos.**

La aplicación cuenta con una base de datos propia para gestionar y almacenar la información. Gracias al sistema de models de Django se ha podido implementar una base de datos completamente funcional utilizando Python.

### **CES1.7: Controlar la calidad y diseñar pruebas en la producción de software.**

Al ser una aplicación para una empresa, se debe garantizar una buena calidad del producto pues una mala solución podría llegar a dañar su reputación. Durante la implementación se han ido probando las funcionalidades con tal de detectar bugs o su mal funcionamiento.

### **CES2.1: Definir y gestionar los requisitos de un sistema software.**

Al ser el único desarrollador de la aplicación, todo el proceso de definición y gestión de requisitos recaía sobre mí. Durante el transcurso de todo el proyecto se ha ido mantenido una comunicación fluida con el equipo de dirección de El Segon Cau para discutir las necesidades, los gustos y las opiniones de las responsables del centro respecto a la web. Con esta información ha sido posible definir y adaptar los requisitos del sistema.

**CES2.2: Diseñar soluciones apropiadas en uno o más dominios de aplicación, utilizando métodos de ingeniería del software que integren aspectos éticos, sociales, legales y económicos.**

La aplicación trabaja en varios dominios, desde el lado del cliente hasta el lado del servidor y de la base de datos. Además, al tratarse de una aplicación web de uso público para una guardería, los aspectos éticos y sociales están muy presentes. Se debe tener mucho cuidado al elegir qué tipo de información se mostrará en la web o qué tipo de imágenes estarán disponibles para el público, sobre todo al tratarse de menores.

**CES3.1: Desarrollar servicios y aplicaciones multimedia.**

El producto final es una aplicación web multimedia. La aplicación ofrece una experiencia multimedia con un gran número de elementos visuales y con partes interactivas. Además es funcional en todo tipo de navegadores y dispositivos.

## 14. Referencias

- [1] Página de El Segon Cau. <http://elsegoncau.blogspot.com/>. Última visita: 9/12/2020
- [2] Blogger. Página principal de Blogger. <https://www.blogger.com/about/>. Última visita: 03/12/2020.
- [3]Scrum. What is scrum? <https://www.scrum.org/resources/what-is-scrum>. Última visita: 03/12/2020
- [4]Taiga. Que es Taiga. <https://tree.taiga.io/support/frequently-asked-questions/who-is-taiga-for/>. Última visita: 03/12/2020.
- [5]Git. Que es Git. <https://git-scm.com/about>. Última visita: 03/12/2020.
- [6]GitHub. Funcionalidades de GitHub. <https://github.com/features>. Última visita: 03/12/2020.
- [7]Terminología Volere para especificación de requisitos. <https://www.volere.org/templates/volere-requirements-specification-template/>. Última visita: 30/12/2020
- [8]Wordpress. Información sobre Wordpress. <https://es.wordpress.org/about/>. Última visita: 31/12/2020
- [9]W3Techs. Porcentaje de uso de sistemas de gestión de contenidos. [https://w3techs.com/technologies/overview/content\\_management](https://w3techs.com/technologies/overview/content_management). Última visita: 31/12/2020
- [10]Vix. Historia de Blogger. <https://www.vix.com/es/btg/tech/2007/05/28/la-historia-de-blogger>. Última visita: 2/1/2021
- [11]Digital55. Que es Ruby on Rails. <https://www.digital55.com/desarrollo-tecnologia/ruby-on-rails-desarrollo-software/>. Última visita: 2/1/2021
- [12] djangoproject. Página principal de Django. <https://www.djangoproject.com/start/overview/>. Última visita: 2/1/2021
- [13]GlassDoor. Sueldos de los roles del proyecto. <https://www.glassdoor.es/Empleo/index.htm>. Última visita: 6/1/2021.
- [14]PyCharm. Precio de una suscripción. <https://www.jetbrains.com/pycharm/buy/#commercial?billing=yearly>. Última visita: 6/1/2021.
- [15]Workspot. Oficina en Barcelona. <https://www.workspot.cat/fix.html>. Última visita: 6/1/2021.
- [16] GDPR en España. Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales. <https://www.boe.es/buscar/doc.php?id=DOUE-L-2016-80807>. Última visita: 09/1/2021



[17] Derechos de imagen. Ley Orgánica de protección civil del derecho al honor, a la intimidad personal y familiar y a la propia imagen.  
<https://www.boe.es/buscar/act.php?id=BOE-A-1982-11196>. Última visita: 09/1/2021