

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN



"APLICACIÓN WEB PARA SERVICIOS DE
CATERING"

TRABAJO FIN DE GRADO

Julio - 2024

AUTOR: Lorena Castillo García
DIRECTOR/ES: Jesús Javier Rodríguez Sala

RESUMEN

El proyecto consiste en el desarrollo de una aplicación web para la gestión de una empresa de catering, cuyo objetivo principal es aumentar la visibilidad y las ventas. La aplicación permite a cualquier usuario registrado reservar un menú compuesto por los productos disponibles, contratar al personal necesario y, si es necesario, reservar un lugar como una finca para realizar el evento. El valor de cada elemento de la reserva, junto con el valor total, se mostrará en el carrito de compras.

La aplicación ofrece una administración visual del carrusel de imágenes inicial, las reseñas de los clientes, los productos, el personal y las fincas, permitiendo la creación, modificación o eliminación de cualquiera de estos elementos. Además, cuenta con un panel de administración para gestionar las reservas de los usuarios.

El desarrollo de la aplicación se ha realizado utilizando Spring Boot como herramienta, conectándose a la base de datos MariaDB, gestionada a través de phpMyAdmin.



AGRADECIMIENTOS

Todas las personas que hicieron posible la realización de este trabajo, haciendo una mención especial a mi tutor Jesuja, gracias por tu guía, dedicación y motivación , siendo una parte fundamental para su finalización.

Y a Vidal, el que desde que comenzó nuestra etapa universitaria me acompaña en el día a día, tanto en lo personal como en lo profesional, sintiendo cada uno de mis logros se sienta como si también fueran suyos.



ÍNDICE GENERAL

1. Introducción	9
1.1 Catering Chapata	9
1.2 Justificación del proyecto	10
1.2.1 Disponer de un sitio web propio	11
1.2.2 Servicio de reservas online	11
1.2.3 Incrementar las ventas con la presencia online	12
1.3 Objetivos	12
1.4 Límites del proyecto	14
2. Antecedentes y estado de la cuestión	15
2.1 Situación actual de la empresa	15
2.2 Herramientas disponibles en el mercado	16
2.2.1 CarterSpot	16
2.2.2 La Seu Catering	19
2.2.3 Catering Ya	24
2.2.4 Resumen	26
2.3 Valoración	30
3. Hipótesis de trabajo	31
3.1.Herramientas de desarrollo	31
3.1.1 Spring Tool Suite 3	31
3.1.2 Arquitectura WAMP	34
3.1.2.1 XAMPP	34
3.1.3 phpMyAdmin	35
3.1.4 Dia	36
3.1.5 GanttProject	37
3.1.6 Maven Project	38
3.2 Framework y librerías	40
3.2.1 Spring boot	40
3.2.2 Bootstrap	44
3.2.3 Librería Java Standard Library	45
3.2.4 Librería JavaServer Pages Standard Tag Library	46
3.2.5 Librería JavaMail API	47
3.2.6 Librería MySQL Connector/J	47
3.3 Arquitectura MVC	47
3.4 Tecnología fronted	49
3.5 Tecnología backend	50
3.5.1 Java	50
3.5.2 JSP (JavaServer Pages)	51
3.5.3 SQL (Structured Query Language)	53

4 Metodología y resultados	55
4.1 Planificación del proyecto	55
4.2 Captura de requisitos	59
4.2.1 Requisitos funcionales	59
4.2.2 Roles de usuario	61
4.2.3 Casos de uso	62
4.3 Diseño	64
4.3.1 Base de datos	64
4.3.2 Diagrama de clase	65
4.3.3 Esbozo	66
4.4 Implementación	66
4.5 Pruebas	74
4.5.1 Pruebas de interfaz de usuario (UI)	74
4.5.2 Pruebas de validación de formularios	75
5 Conclusiones y trabajo futuro	78
5.1 Conclusiones	78
5.2 Posibles desarrollos futuros	79
6 Bibliografía	81
7 Anexo I. Casos de uso	87
7 Anexo I.1. Usuario no registrado	88
7 Anexo I.2. Usuario registrado	100
7 Anexo I.3. Administrador	113
7 Anexo II. Detalle del diagrama de clases	139

ÍNDICE DE TABLAS

Tabla 2.1 Tabla comparativa de características (I)	27
Tabla 2.2 Tabla comparativa de características (II)	28
Tabla 2.3 Tabla comparativa de características (III)	29
Tabla 3.1 Directivas de JSP	53
Tabla 4.1 Rol usuario no registrado	61
Tabla 4.2 Rol usuario registrado	61
Tabla 4.3 Rol administrador	62
Tabla AI.1 Caso de uso 1	89
Tabla AI.2 Caso de uso 2	90
Tabla AI.3 Caso de uso 3	91
Tabla AI.4 Caso de uso 4	92
Tabla AI.5 Caso de uso 5	93
Tabla AI.6 Caso de uso 6	94
Tabla AI.7 Caso de uso 7	95
Tabla AI.8 Caso de uso 8	96
Tabla AI.9 Caso de uso 9	97
Tabla AI.10 Caso de uso 10	97
Tabla AI.11 Caso de uso 11	98
Tabla AI.12 Caso de uso 12	99
Tabla AI.13 Caso de uso 13	101
Tabla AI.14 Caso de uso 14	102
Tabla AI.15 Caso de uso 15	103
Tabla AI.16 Caso de uso 16	104
Tabla AI.17 Caso de uso 17	105
Tabla AI.18 Caso de uso 18	106
Tabla AI.19 Caso de uso 19	107
Tabla AI.20 Caso de uso 20	108
Tabla AI.21 Caso de uso 21	108
Tabla AI.22 Caso de uso 22	109
Tabla AI.23 Caso de uso 23	110
Tabla AI.24 Caso de uso 24	110
Tabla AI.25 Caso de uso 25	111
Tabla AI.26 Caso de uso 26	112
Tabla AI.27 Caso de uso 27	114
Tabla AI.28 Caso de uso 28	115
Tabla AI.29 Caso de uso 29	116
Tabla AI.30 Caso de uso 30	117
Tabla AI.31 Caso de uso 31	118

Tabla AI.32 Caso de uso 32	119
Tabla AI.33 Caso de uso 33	120
Tabla AI.34 Caso de uso 34	121
Tabla AI.35 Caso de uso 35	122
Tabla AI.36 Caso de uso 36	123
Tabla AI.37 Caso de uso 37	124
Tabla AI.38 Caso de uso 38	125
Tabla AI.39 Caso de uso 39	126
Tabla AI.40 Caso de uso 40	127
Tabla AI.41 Caso de uso 41	128
Tabla AI.42 Caso de uso 42	129
Tabla AI.43 Caso de uso 43	130
Tabla AI.44 Caso de uso 44	131
Tabla AI.45 Caso de uso 45	132
Tabla AI.46 Caso de uso 46	133
Tabla AI.47 Caso de uso 47	133
Tabla AI.48 Caso de uso 48	134
Tabla AI.49 Caso de uso 49	135
Tabla AI.50 Caso de uso 50	136
Tabla AI.51 Caso de uso 51	137
Tabla AI.52 Caso de uso 52	138



ÍNDICE DE FIGURAS

Figura 2.1 Home page CaterSpot	17
Figura 2.2 Home page CarterSpot	18
Figura 2.3 Búsqueda en CaterSpot y Filtro búsqueda parte 1 y 2	19
Figura 2.4 Menú hamburguesa	20
Figura 2.5 Página principal parte 1 y 2	21
Figura 2.6 Formulario de contacto y Página de menús	21
Figura 2.7 Archivo coffe break	22
Figura 2.8 Página de espacio y Página Pinterest	22
Figura 2.9 Página de Catering Eventos	23
Figura 2.10 Página de Servicios complementarios	23
Figura 2.11 Imagen del menú hamburguesa	24
Figura 2.12 Pagina finca para bodas y Opciones de ubicaciones	25
Figura 2.13 Página de gastronomía	25
Figura 2.14 Reserva de cocktail, Detalles de facturación y Carrito	26
Figura 3.1 Logo Spring Tool Suite	32
Figura 3.2 Spring Tool Suite 3 (versión 3.9.9.)	33
Figura 3.3 Panel de control XAMPP	35
Figura 3.4 Panel de control phpMyAdmin	36
Figura 3.5 Interfaz de la herramienta Dia	37
Figura 3.6 Interfaz de la GanttProject	38
Figura 3.7 Estructura de un proyecto con Maven	38
Figura 3.8 Archivo pom.xml	39
Figura 3.9 Módulos Spring	41
Figura 3.10 Archivo root-context.xml	43
Figura 3.11 Archivo PersonalControler.java	44
Figura 3.12 Patrón MVC	48
Figura 3.13 Ejemplo implantación MVC con Spring Boot	48
Figura 3.14: Imagen java.	51
Figura 4.1: Ciclo de vida en cascada	56
Figura 4.2: Planificación de tareas y fechas	58
Figura 4.3: Planificación de tareas línea temporal	59
Figura 4.4: Diagrama jerárquico de los usuarios	62
Figura 4.5: Diagrama casos de uso del usuario no registrado	62
Figura 4.6: Diagrama casos de uso del usuario registrado	63
Figura 4.7: Diagrama casos de uso del administrador	63
Figura 4.8: Diseño base de datos	65
Figura 4.9: Diagrama de clases	65
Figura 4.10: Primer diseño de interfaz de la aplicación.	66

Figura 4.11: Página principal.	67
Figura 4.12: Página de productos	67
Figura 4.13: Página de personal.	68
Figura 4.14: Página de finca y página de contacto.	69
Figura 4.15: Distintos header	70
Figura 4.16: Carrito	71
Figura 4.17: Panel de administrador.	72
Figura 4.18: Panel de reservas	73
Figura 4.19: Panel de usuarios	74
Figura 4.20: Pop up control de unidades	75
Figura 4.21: Pop up producto añadido y confirmación de borrado	75
Figura 4.22: Validación del formulario de registro	76
Figura 4.23: Validación del formulario de login.	76
Figura 4.24: Validación del campo email.	77
Figura 4.25: Anotación CustomEmail.	77
Figura 4.26: Anotaciones predefinidas para números	77
Figura AI.1: Diagrama jerárquico de los usuarios	87
Figura AI.2: Diagrama casos de uso del usuario no registrado	88
Figura AI.3: Diagrama casos de uso del usuario registrado	100
Figura AI.4: Diagrama casos de uso del administrador	113
Figura AII.1: Diagrama de clases (esquema)	139

Capítulo 1

Introducción

1.1.- CATERING CHAPATA

En este trabajo de final de grado abordaremos un problema de digitalización en el campo de la hostelería. En estos momentos, la presencia en Internet se ha convertido en un componente indispensable para el éxito de cualquier negocio, y la industria de la hostelería no es una excepción.

Si bien es cierto que muchas empresas del sector recurren a plataformas de redes sociales como Facebook, Twitter o Instagram para promocionarse de manera económica y, en cierta medida, efectiva, la creación de una página web propia puede marcar la diferencia significativa entre destacar o pasar desapercibido en un mercado saturado.

El enfoque del presente trabajo se va a centrar en un caso hipotético de empresa de catering a la que denominaremos “*Cátering Chapata*”, esta empresa estará especializada en la

preparación y servicio de desayunos, comidas y cenas para eventos y celebraciones. Chapata no solo ofrece una simple oferta gastronómica, sino que también proporciona una amplia gama de servicios complementarios que incluyen la contratación de personal especializado como camareros, cocteleros y jamoneros, así como la gestión integral de la logística del evento, cubriendo incluso aspectos como la limpieza y el transporte. Asimismo, la compañía brinda la comodidad de reservar espacios para eventos, ofreciendo una solución completa a las necesidades de sus clientes.

En este contexto, la creación de una plataforma web dedicada para Catering Chapata se presenta como una estrategia fundamental para potenciar su presencia en el mercado y mejorar la experiencia del cliente. Una presencia en Internet sólida no solo proporciona una ventana a los valores y la identidad de la empresa, sino que también facilita una mayor interacción con los clientes, permitiendo el feedback directo y la posibilidad de ajustar continuamente los servicios ofrecidos para satisfacer las necesidades de un mercado que siempre está en constante evolución. Asimismo, una plataforma web ofrece credibilidad y confianza a los potenciales clientes, estableciendo así un vínculo más sólido y duradero con la marca.

Por todo esto, el desarrollo de la webapp para Catering Chapata, se espera que no solo sirva como un escaparate digital para sus productos y servicios, sino, que también funcione como una herramienta efectiva para la gestión de todo tipo de eventos, así como para la interacción con los clientes.

La plataforma permitirá a los usuarios explorar el catálogo de servicios ofrecidos por Chapata y realizar reservas en línea. Además, se incorporarán funcionalidades adicionales, como la integración de un sistema de gestión de personal y la posibilidad de personalizar menús y servicios según las preferencias del cliente, con el objetivo de llevar a Catering Chapata al siguiente nivel en términos de digitalización y proporcionar una solución completa y personalizada para las necesidades de sus clientes en un mercado cada vez más competitivo y exigente.

1.2.- JUSTIFICACIÓN DEL PROYECTO

El impulso principal detrás de este trabajo reside en la evolución de las dinámicas del mercado. En plena era tecnológica, la manera de ofrecer y comercializar servicios de catering (o de cualquier otro tipo) ha experimentado un cambio radical. Ya no basta con depender únicamente de la presencia física de un establecimiento, los carteles o los rótulos para atraer clientes. En la actualidad, antes de tomar una decisión sobre dónde adquirir un producto o contratar un servicio, la primera acción que realiza casi todo el mundo es buscar información en Internet.

Por ello, la presencia en Internet es crucial para el éxito de cualquier empresa. Aproximadamente el 40% de los usuarios que buscan servicios en línea eligen hacer sus reservas a través de páginas web especializadas, superando las reservas realizadas por teléfono, según estimaciones. La reserva online es en gran medida responsable de esta tendencia debido a la conveniencia que ofrece. Cuando se automatiza el proceso de reserva, el personal del establecimiento reduce significativamente el tiempo dedicado a gestionar estas reservas, lo que les permite enfocarse en otras tareas cruciales. También, la flexibilidad de poder reservar en cualquier momento, sin importar el horario de atención del lugar, se ajusta perfectamente a las necesidades de los clientes actuales [1].

1.2.1.- Disponer de un sitio web propio

Para mejorar la competitividad, los propietarios manejarán el servicio de catering directamente desde un sitio web, teniendo el dominio completo sobre la presentación de la información. Ellos serán quienes elijan qué publicar, cómo y cuándo. Este enfoque asegurará que la oferta permanezca actualizada, lo que permitirá brindar el mejor servicio posible y ofrecer a los clientes una amplia gama de opciones.

Con ello se consigue dar una imagen de profesionalidad, disponiendo también de un contacto directo con el cliente sin necesidad de intermediarios. Esta comunicación directa es de suma importancia, especialmente considerando la creciente preocupación por los alérgenos. Es cada vez más común que los clientes requieran modificaciones en los platos o en el lugar del evento debido a alergias alimentarias, como el gluten o la lactosa, por ejemplo.

Además, en la plataforma web que se va a desarrollar, los precios de los productos y servicios estarán claramente visibles en todo momento, acompañados de imágenes atractivas. Como se suele decir, una imagen vale más que mil palabras; en este caso, fotografías de los platos exquisitamente presentados aumentarán significativamente el atractivo para cualquier cliente potencial, despertando su interés y apetito.

1.2.2.- Servicio de reservas online

Con el objetivo de optimizar la gestión de reservas, se va a implementar un sistema online personalizado que permitirá a los usuarios realizar reservas de manera eficiente. A través de la plataforma, se presentarán diversas opciones para reservar servicios tanto de comida, personal, como de espacios para eventos, adaptándose a las fechas específicas de cada evento.

De esta forma para el usuario será muy cómoda la reserva en la web, mientras que para el dueño del establecimiento le permitirá aceptar clientes sin necesidad de estar pendiente del teléfono y consultando la base de datos para saber qué productos o personal habría disponibles para la persona que llama pidiendo información.

1.2.3.- Incrementar las ventas con la presencia online

Aunque las razones anteriores son una base sólida para la creación de un servicio web como el que se está describiendo, el impulso principal detrás de una iniciativa de este tipo es económico, buscando atraer a una cartera de clientes más amplia y diversa.

Se puede mantener a la empresa a la par con otros competidores gracias a una presencia online, lo que asegura que las ofertas y servicios estén siempre actualizados y alineados con las expectativas del mercado. Todo esto contribuye a la fidelización de los clientes existentes y a mejorar la experiencia del usuario con esta constante actualización. Al establecer una sólida cartera de clientes satisfechos, no solo se garantiza un crecimiento constante, sino que también se sientan las bases para futuras expansiones comerciales.

Asimismo, brinda la posibilidad de utilizar las tendencias nacientes y las nuevas oportunidades del mercado, lo que puede resultar en una mayor visibilidad y alcance para esta empresa. En síntesis, una estrategia digital no solo aumentará las ventas, sino que también refuerza una posición competitiva que busca el éxito a largo plazo.

1.3.- OBJETIVOS

Como objetivo personal, me planteo aprender un nuevo lenguaje de programación con nuevas tecnologías. Para este proyecto se va a usar Spring Framework, ya que actualmente tiene una alta demanda en el mercado laboral.

Spring fue creado por Rod Johnson en 2003, nació con la idea principal de que sirviera como una plataforma para desarrollar en Java de código abierto y gracias a esto su uso comenzó a extenderse hasta convertirse en el framework más popular para Java en un ámbito empresarial, para crear código de alto rendimiento, liviano y reutilizable.

Spring pretende reducir al mínimo el número de pasos que un desarrollador debe dar en la configuración inicial del proyecto antes de ponerse a trabajar en la parte dura del proyecto centrandose en lo importante. La idea es clara: no perder el tiempo (y el dinero) en hacer las mismas cosas una y otra vez [2].

Además se usará Spring Boot que es una extensión, un suite, pre-configurado para ejecutar Spring utilizando las mejores prácticas sin por ello perder flexibilidad. En definitiva, Spring Boot se encarga de la mayor parte del trabajo pesado que supone levantar una aplicación Spring [3].

En líneas generales, se creará un proyecto pre-configurado mediante Spring Boot con Maven, conectado al servidor Pivotal que está incluido en Spring. Esto irá conectado con la base de datos para toda la gestión de usuarios y para la parte de la interfaz se usará Bootstrap, que es una biblioteca multiplataforma que contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end [4].

El objetivo principal de este trabajo es potenciar la visibilidad de un negocio de catering para incrementar sus ventas. Para ello se quiere desarrollar una aplicación multiplataforma, que funcione en cualquier dispositivo (PC, portátil, tablet o móvil) y sistema operativo (Windows, Mac, Android, iOS, etc.), por lo que se optará por diseñar un servicio web que funcione en cualquier navegador web. Aunque actualmente su diseño estará enfocado al navegador de un ordenador, estará preparado para utilizarse en dispositivos móviles, ya que hoy en día los usuarios utilizan más el móvil que el ordenador.

Por otro lado, el propósito profesional de desarrollar esta aplicación es amplio, enfocado en mejorar tanto la experiencia del cliente como la eficiencia operativa del negocio.

Se busca mejorar la accesibilidad y comodidad para los clientes al crear una plataforma fácil de usar y accesible desde cualquier dispositivo con conexión a Internet. De esta manera, se promoverá un aumento en las ventas y la lealtad de los clientes al brindar una experiencia satisfactoria y personalizada a través de reservas rápidas y convenientes para ellos. También se busca mejorar la gestión de reservas, facilitando al negocio controlar la disponibilidad de servicios, personal y espacios para eventos. Mejorando de esta forma la eficiencia operativa, esta acción contribuirá a mantener actualizado y competitivo el negocio en un entorno digital en constante evolución.

Además, se pretende aumentar las ventas y retener clientes atrayendo a una cartera de clientes más amplia y facilitando la reserva de servicios. Se diseñará la aplicación para que se adapte a las últimas tendencias tecnológicas y necesidades del mercado, asegurando que el negocio permanezca relevante y competitivo.

Por último, la aplicación tiene que establecer los cimientos para futuras expansiones comerciales al ofrecer una plataforma escalable y flexible capaz de adaptarse al crecimiento del negocio y a la inclusión de nuevos servicios. A medida que evolucionen las necesidades y demandas del mercado, esto facilitará la expansión del negocio.


1.4.- LÍMITES DEL PROYECTO

Hasta el momento, se ha delineado de manera general la funcionalidad principal de la aplicación. Sin embargo, con miras a mejorar la funcionalidad y flexibilidad de la plataforma, se contempla la posibilidad de llevar a cabo varias expansiones de roles en el futuro. Estas podrían incluir la introducción de un rol de diseñador para abordar la maquetación del sitio web, la incorporación de un rol autónomo para ofrecer servicios en el personal a contratar así como la creación de un rol de propietario a facilitar la oferta de propiedades, como fincas o espacios para eventos.



Capítulo 2

Antecedentes y estado de la cuestión



2.1.- SITUACIÓN ACTUAL DE LA EMPRESA

En la actualidad, el sector de los servicios de catering se enfrenta a una serie de desafíos significativos en cuanto a la gestión y promoción de servicio que, en muchos casos, resulta poco práctico a pesar de la cantidad en una variedad de eventos y celebraciones que se realizan, que pueden ser desde bodas hasta conferencias corporativas. A menudo, los servicios de catering se ven limitados por ciertas circunstancias que pueden obstaculizar su eficiencia y capacidad para satisfacer las necesidades cambiantes de los clientes.

Uno de los desafíos más prominentes es la gestión operativa. Coordinar la preparación de alimentos, el personal necesario, la logística de entrega o el lugar específico de cada evento puede resultar complejo y exigente. Esta complejidad aumenta aún más cuando se consideran las expectativas cada vez mayores de los clientes en términos de calidad, personalización y presentación.

Además, la promoción y comercialización efectiva de los servicios de catering también plantea desafíos significativos. En un mundo digitalizado, donde las redes sociales y las plataformas en Internet juegan un papel crucial en la toma de decisiones de los consumidores, los proveedores de catering deben destacarse entre la multitud y llegar a su público objetivo de manera efectiva. Esto requiere estrategias de marketing innovadoras y una presencia en Internet sólida y atractiva.

Otro desafío clave es la adaptación a las tendencias y preferencias cambiantes del mercado. Con la creciente conciencia sobre la sostenibilidad, la salud y la diversidad alimentaria, los servicios de catering deben ser capaces de ofrecer opciones que satisfagan estas demandas emergentes sin comprometer la calidad o la experiencia del cliente.^[5]

A pesar de estos desafíos, también existen numerosas oportunidades para el crecimiento y la innovación en el sector de los servicios de catering. La demanda de experiencias gastronómicas únicas y personalizadas sigue siendo alta, lo que brinda a los proveedores la oportunidad de diferenciarse a través de la creatividad y la excelencia en el servicio. Además, la tecnología ofrece herramientas poderosas para mejorar la eficiencia operativa, la comunicación con los clientes y la gestión de eventos.

En conclusión, puede decirse que la situación actual de los servicios de catering es una mezcla de desafíos y oportunidades. Si bien existen obstáculos que deben superarse, como la gestión operativa compleja y la competencia creciente, también hay espacio para la innovación y el crecimiento, especialmente para aquellos que puedan adaptarse rápidamente a las necesidades y expectativas cambiantes de los clientes. ^[6]

2.2.- HERRAMIENTAS DISPONIBLES EN EL MERCADO

A continuación se van a describir algunas aplicaciones web que ya existen en el mercado, y que ofrecen servicios de catering. Todas estas herramientas se corresponden con empresas reales del sector. Sobre las capturas de pantalla que se muestran en los apartados siguientes, mencionar que, si bien todas las empresas disponen de webs adaptativas, se ha optado, en la mayoría de los casos, por mostrar la versión para dispositivo móvil.

2.2.1.- CarterSpot

CaterSpot ^[7] es una plataforma que opera en ciudades determinadas con más de 500 proveedores, ofrece una amplia variedad de opciones para todos los gustos y necesidades.

Desde catering corporativo hasta almuerzos individuales para empleados. Además, su enfoque en la gestión de despensas o de provisiones y cafeterías hace que sea una solución integral para empresas.

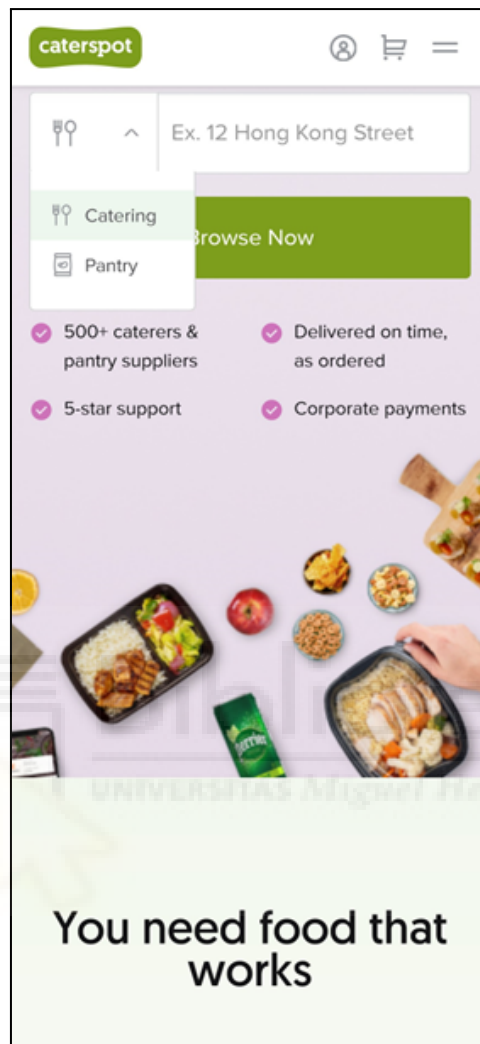


Figura 2.1: Home page CaterSpot

No dispone de servicio en España, y es similar a Uber Eats o Glovo. Las empresas con catering ponen su oferta y el usuario selecciona lo que desea en base a los filtros seleccionados. Los servicios que ofrece son los siguiente:

- Pedidos en línea fáciles: Dispone de más de 500 proveedores para realizar pedidos en línea.
- Catering corporativo: Comida para reuniones y eventos, o lleva el restaurante a la oficina.
- Almuerzos para empleados: Se trata de comidas individuales de restaurantes locales populares. Dispone de una opción de pedido grupal para que los equipos elijan sus propias comidas.

- Despensa de oficina: Mantiene bien abastecida la despensa de una oficina con snacks, bebidas y suministros de comestibles.
- Soluciones gestionadas para cafetería y despensa de oficina: Manejo de pedidos y gestión de inventario para cafeterías y despensas.
- Planes de comidas para el personal: Tanto si la empresa tiene 50 como si tiene 5,000 empleados, sus programas de comidas pueden satisfacer todas las necesidades y presupuestos.
- Gestión de despensa: Se encargan de la adquisición, el pedido y la optimización del presupuesto y el stock.
- Gestión de cafetería: Una solución empresarial personalizada para los programas de alimentación de una oficina.

La interfaz es sencilla e intuitiva pero muy cargada . Cuando se abre la web lo primero que aparece es una “*home page*” en la que se solicita la dirección de donde se va a realizar el servicio. También da la opción de si se desea una comida/cena/desayuno o un picoteo en el desplegable del tenedor. Arriba a la derecha tenemos un menú hamburguesa.

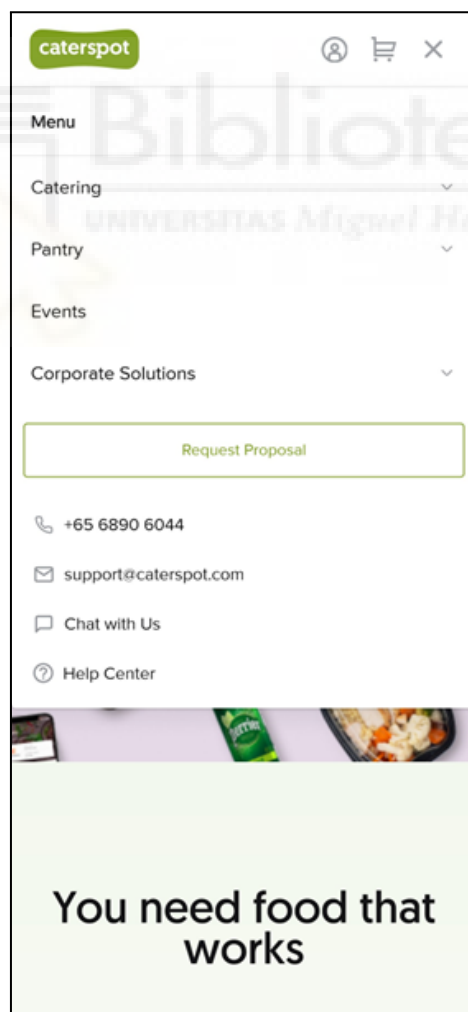


Figura 2.2: Home page CaterSpot

Abriendo el menú hamburguesa se muestran tanto opciones de catering, como de despensa y como de eventos.

Seleccionando la opción de Catering para buscar por catering, por menú, por plato etc. Además a la derecha del buscador hay un filtro con multitud de opciones. A modo de prueba se ha realizado una búsqueda por desayuno:

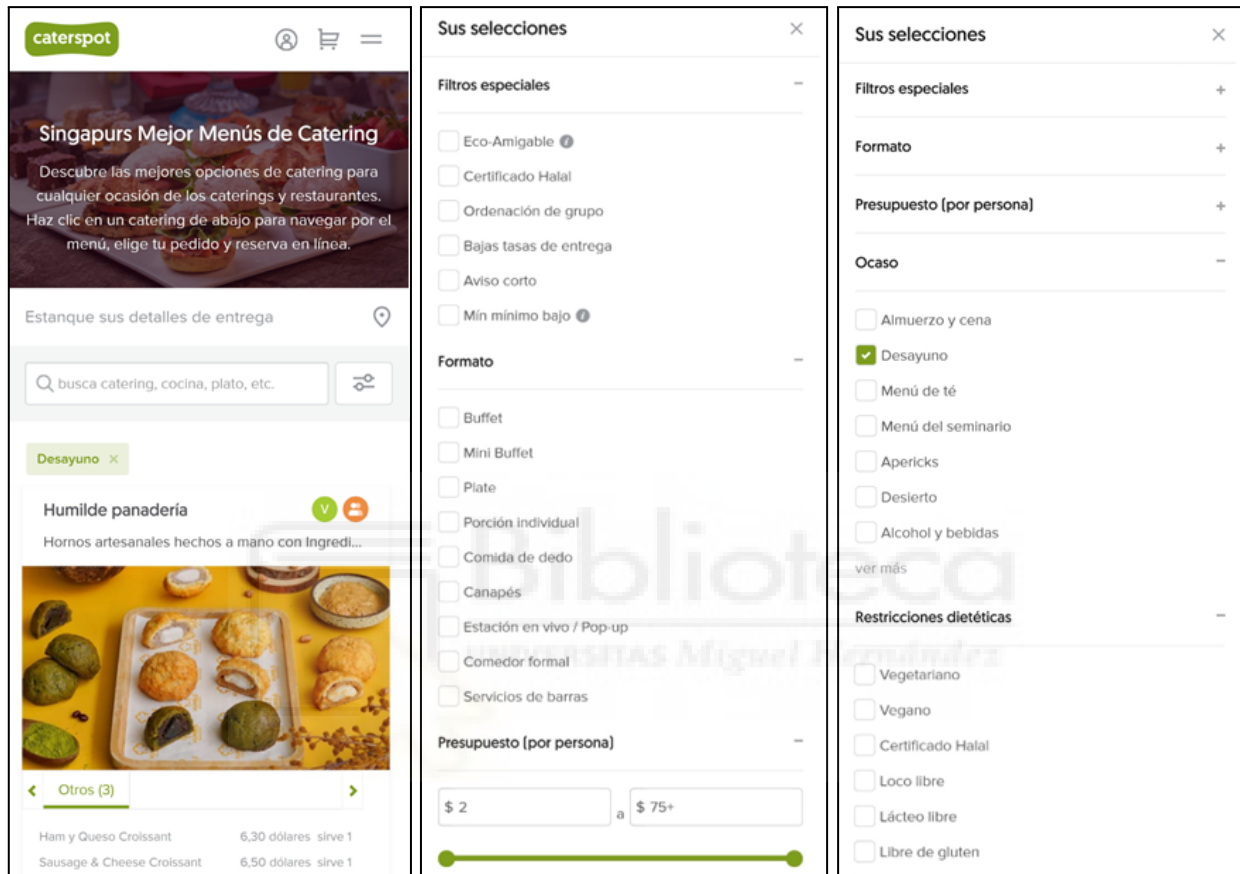


Figura 2.3: Búsqueda en CaterSpot y Filtro búsqueda parte 1 y 2

No se ha podido realizar ninguna compra ni intento de ella porque no operan en España.

2.2.2.- La Seu Catering

La Seu Catering [8] es una empresa de catering y hostelería cuya sede se encuentra en Valencia. Tiene más de 15 años de experiencia ofreciendo este tipo de servicio. En su página web se pueden ver algunas de sus opciones de menús, personal y espacios donde celebrar eventos como se muestra a continuación.

Al abrir la web se observa un menú de navegación superior permite acceder fácilmente a diferentes secciones, como Quienes somos, Menús, Espacios, Catering Eventos, gastronomía y Contacto.

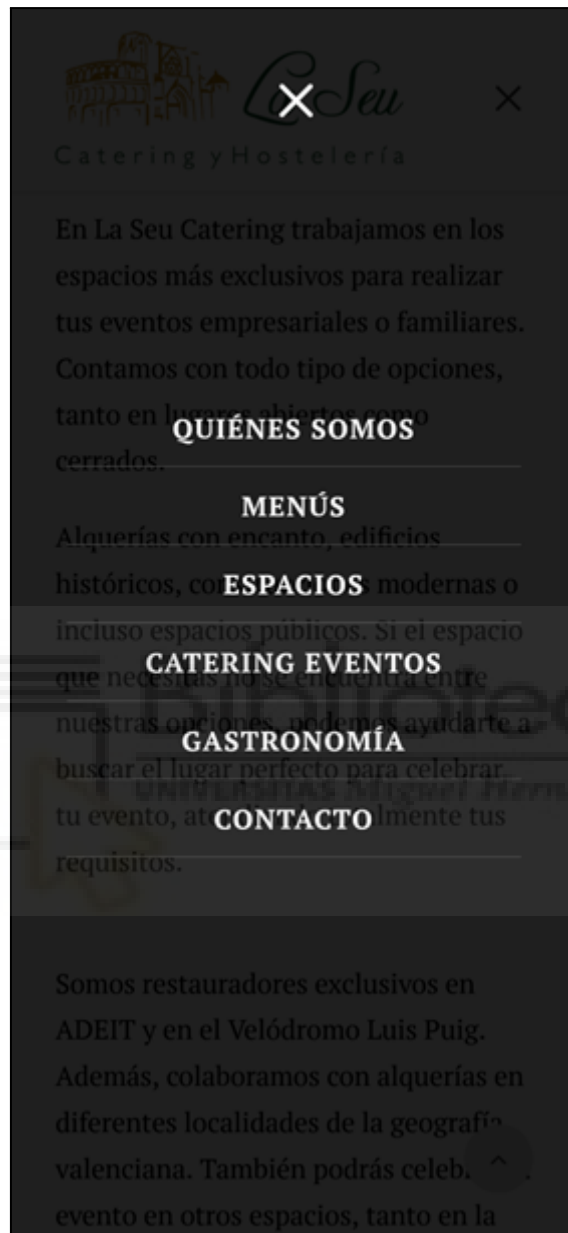


Figura 2.4: Menú hamburguesa

La página de inicio da la bienvenida a los visitantes mediante un un carrusel de imágenes y destaca la experiencia de más de 15 años en el mundo del catering y la restauración. Continúan ofreciendo una variedad de menús para diferentes ocasiones, como pueden ser Desayunos de Trabajo, Congresos o Celebraciones Familiares entre las que destacan bodas íntimas, cumpleaños y aniversarios.

Finalmente disponemos de un contacto y un formulario para solicitar presupuesto. El resto de páginas son todas muy similares, son imágenes que contienen enlaces o archivos

descargables. La página de Menú, muestra bastantes opciones de menús con enlaces a documentos pdf para descargar, con la información de cada menú.

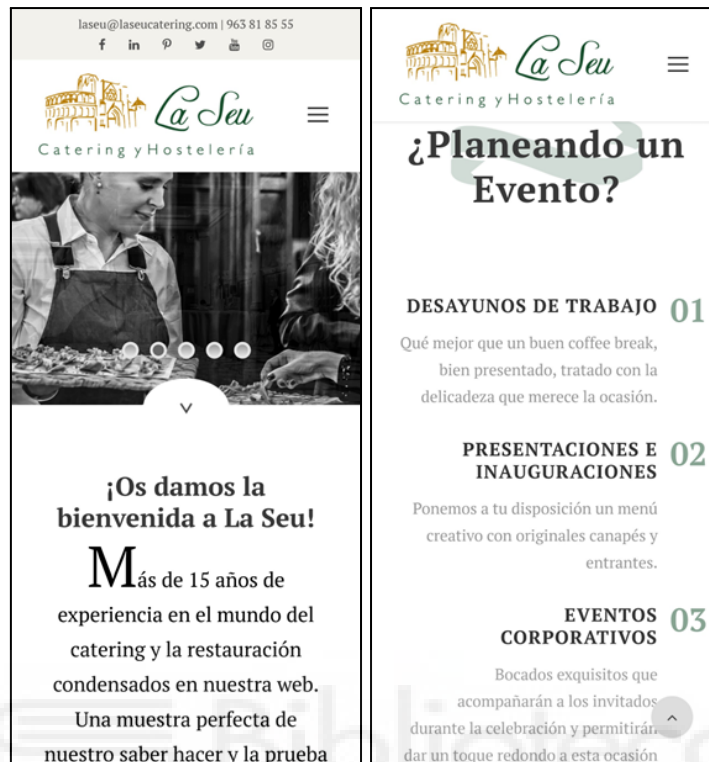


Figura 2.5: Página principal parte 1 y 2



Figura 2.6: Formulario de contacto y Página de menús



Catering y Hostelería

¿QUÉ ESTA CONTRATANDO?

Un coffee break es un servicio de restauración que se realiza como tiempo de descanso entre unas actividades (ponencias, debates, cursos, reuniones).

La duración aproximada de este servicio es de unos 30 minutos.

Para este tipo de catering habilitamos un espacio para el servicio de cafés e infusiones mediante termos o cafeteras exprés que nos permiten realizar el café en el momento del servicio. La comida dulce y/o salada se deja en las mesas de apoyo donde, además, los comensales, tienen a su disposición servilletas.

En los coffee break dulces-salados o salados, parte de la comida y la bebida puede ser pasada por el personal de servicio (dependiendo de las características del lugar y del menú).

Para servicios a primera hora de la mañana les recomendamos los coffee break básicos (en sus dos variantes de zumo de naranja envasado y natural, recién exprimido). Para servicios a última hora de la mañana, los dulce-salados o salados. Si usted piensa realizarlo a primera hora de la tarde, las mejores opciones son, nuevamente las del coffee break básico o la del clásico. Para recesos a media tarde, y especialmente en verano, la mejor opción es el coffee break valenciano con horchata.

Debido al Protocolo COVID no podemos dejar la comida dispuesta encima de las mesas al alcance de todos los comensales, por este motivo la comida se sirve desde una mesa buffet atendida por nuestro personal en platos individuales.

NUESTRA PROPUESTA GASTRONÓMICA

BÁSICO "A"

Mini Croissant Francés de Mantequilla (Alérgenos nº1, 3, 7, 6, 8, 12)
Mini Ensaimadas (Alérgenos nº1, 3, 4, 5, 7, 8)
Mini Sneken Danés con Crema y Pasas (Alérgenos nº1, 3, 6, 7, 8, 12)
 * * *

Café, Descafeinados e Infusiones.
 * * *

Zumos de Naranja, Melocotón y Piña.
Agua Mineral



BÁSICO "B"

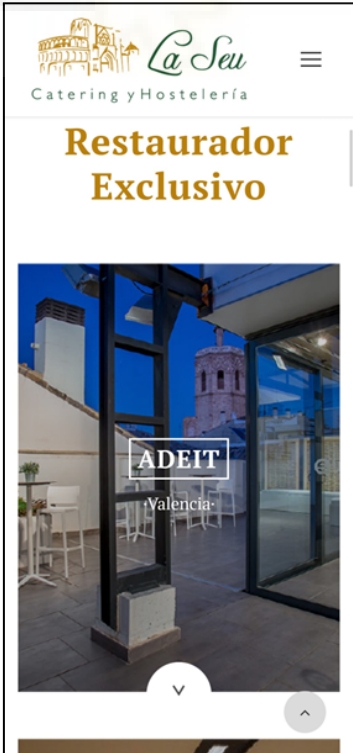
Mini Croissant Francés de Mantequilla (Alérgenos nº1, 3, 7, 6, 8, 12)
Mini Ensaimadas (Alérgenos nº1, 3, 4, 5, 7, 8)
Mini Sneken Danés con Crema y Pasas (Alérgenos nº1, 3, 6, 7, 8, 12)
Chuchús de Crema (Alérgenos nº1, 3, 7, 8)
Berlinas de Azúcar (Alérgenos nº1, 3, 5, 6, 7, 8)
Tradicional Coca de Llanda (Alérgenos nº1, 3, 5, 6, 7, 8)
 * * *

Café, Descafeinados e Infusiones.
 * * *

Zumos de Naranja, Melocotón y Piña.
Agua Mineral

Figura 2.7: Archivo coffee break

En el apartado de espacios se muestra una imagen por cada lugar disponible, al clicar en una de ellas se enlaza con la web de Pinterest. Cabe destacar que en todas las opciones del menú navegable está el formulario para pedir presupuesto. Es el mismo formulario en todas las páginas, el mismo que se muestra en la página de contacto.



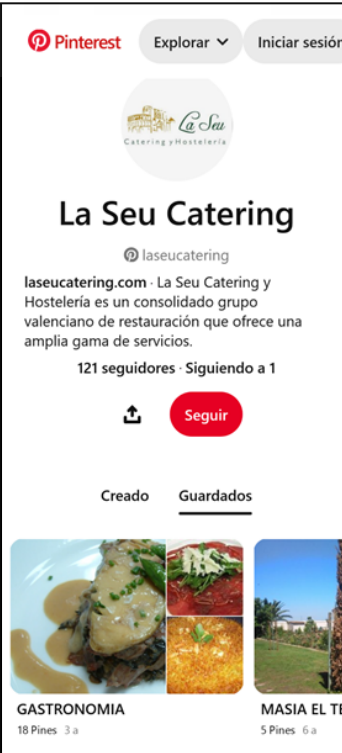


Figura 2.8: Página de espacio y Página Pinterest

Por último, aunque parece disponer de servicios complementarios parece que está por desarrollar ya que algunas de las opciones llevan a un post con el texto por defecto “*Lorem Ipsum*”, que es un fragmento de texto falso que se utiliza en diseño gráfico y maquetación cuyo propósito es llenar espacios de contenido antes de tener el texto definitivo.



Figura 2.9: Página de Catering Eventos



Figura 2.10: Página de Servicios complementarios

La web aparte de visualizar contenido y solicitar presupuesto no dispone de nada más.

2.2.3.- Catering Ya

Catering Ya [9] es una empresa con más de 40 años de experiencia en servicios de catering para eventos en Alicante y Murcia. Se especializan en proporcionar servicios de catering para bodas, comuniones y otros eventos.

Lo primero que muestra la página principal es un carrousel de imágenes y un menú hamburguesa con las siguientes opciones.

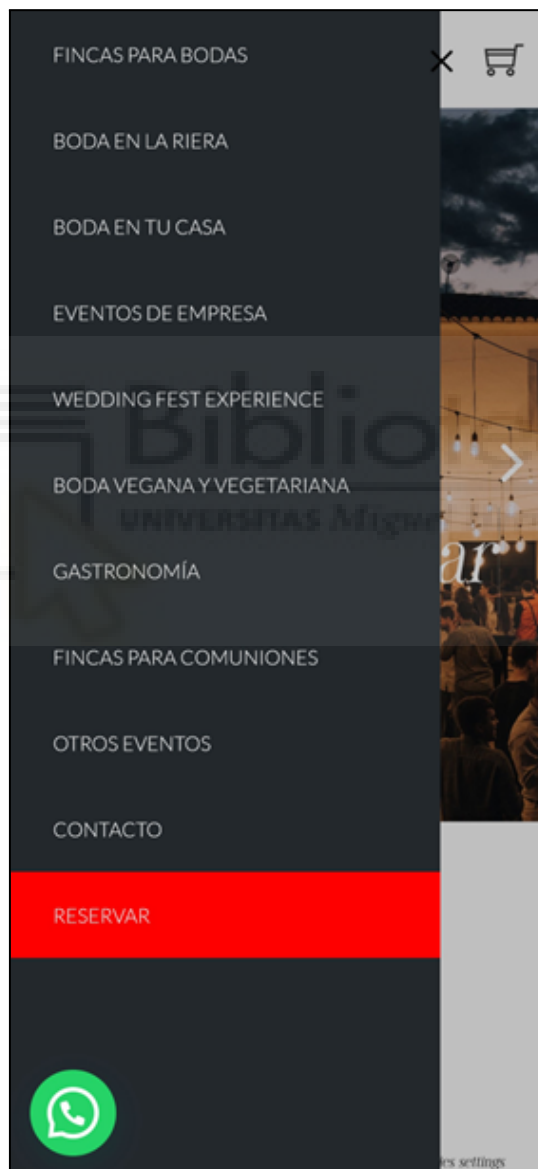


Figura 2.11: Imagen del menú hamburguesa

En la opción de fincas para bodas aparece la opción de elegir el lugar, si se desea cubierto o no y su aislamiento.



Figura 2.12: Pagina finca para bodas y Opciones de ubicaciones

En la sección de gastronomía solo aparecen fotos con la descripción del plato.



Figura 2.13: Página de gastronomía

En la página de reserva hay opciones definidas de reserva, no termina de estar claro que se está reservando en concreto, porque no se solicita ni menú, ni finca, ni un día para la realización del evento, pero si piden 300€.



Figura 2.14: Reserva de cocktail, Detalles de facturación y Carrito

Finalmente aunque no hay una opción de login o registro, en la página de carrito sí se piden unos datos de facturación que se deben rellenar obligatoriamente, de lo contrario no permite realizar el pago.

2.2.4.- Resumen

A continuación, a modo de resumen se muestra una tabla comparativa de las características de cada una de las herramientas vistas con respecto a la aplicación a realizar en este proyecto.

(ver en las páginas siguientes)

Tabla 2.1: Tabla comparativa de características (I)

	CaterSpot	Chapata
Diseño y Usabilidad del Sitio Web		
La interfaz del sitio web es intuitiva y fácil de navegar	Si	Si
El diseño es atractivo y profesional	Si	Si
La información está organizada de manera clara y accesible	Si	Si
Módulo de Inicio		
Dispone de elementos interactivos o multimedia	Si	Si
El tipo de contenido se muestra aquí deja claro de qué trata la web	Si	Si
Módulo de Productos		
Se pueden categorizar los productos de alguna manera	Si	Si
Se muestra para cada producto su precio	Si	Si
Se muestra descripción del producto	Si	Si
Módulo de Personal		
Se puede seleccionar el personal	No	Si
Se ven las tarifas del personal a contratar	No	Si
Módulo de Finca		
Dispone de detalles específicos, como ubicación, tamaño, o características	No	Si
Dispone de imágenes de la finca	No	Si
Módulo de Contacto		
Se proporciona un formulario de contacto	Si	Si
Módulo de Login		
Se dispone de registro y login en la plataforma	Si	Si
Módulo de Carrito		
Dispone de control de unidades de productos	Si	Si
Dispone de opciones de pago	Si	No
Módulo de Reservas		
Se dispone de un histórico de las reservas realizadas y su estatus	Si	Si
Módulo de Perfil		
El usuario puede gestionar su perfil personal	Si	Si

Tabla 2.2: Tabla comparativa de características (II)

	La Seu	Chapata
Diseño y Usabilidad del Sitio Web		
La interfaz del sitio web es intuitiva y fácil de navegar	Si	Si
El diseño es atractivo y profesional	Si	Si
La información está organizada de manera clara y accesible	No	Si
Módulo de Inicio		
Dispone de elementos interactivos o multimedia	Si	Si
El tipo de contenido se muestra aquí deja claro de qué trata la web	Si	Si
Módulo de Productos		
Se pueden categorizar los productos de alguna manera	No	Si
Se muestra para cada producto su precio	No	Si
Se muestra descripción del producto	Si	Si
Módulo de Personal		
Se puede seleccionar el personal	No	Si
Se ven las tarifas del personal a contratar	No	Si
Módulo de Finca		
Dispone de detalles específicos, como ubicación, tamaño, o características	No	Si
Dispone de imágenes de la finca	Si	Si
Módulo de Contacto		
Se proporciona un formulario de contacto	Si	Si
Módulo de Login		
Se dispone de registro y login en la plataforma	No	Si
Módulo de Carrito		
Dispone de control de unidades de productos	No	Si
Dispone de opciones de pago	No	No
Módulo de Reservas		
Se dispone de un histórico de las reservas realizadas y su estatus	No	Si
Módulo de Perfil		
El usuario puede gestionar su perfil personal	No	Si

Tabla 2.3: Tabla comparativa de características (III)

	Ya	Chapata
Diseño y Usabilidad del Sitio Web		
La interfaz del sitio web es intuitiva y fácil de navegar	No	Si
El diseño es atractivo y profesional	No	Si
La información está organizada de manera clara y accesible	No	Si
Módulo de Inicio		
Dispone de elementos interactivos o multimedia	Si	Si
El tipo de contenido se muestra aquí deja claro de qué trata la web	Si	Si
Módulo de Productos		
Se pueden categorizar los productos de alguna manera	No	Si
Se muestra para cada producto su precio	No	Si
Se muestra descripción del producto	No	Si
Módulo de Personal		
Se puede seleccionar el personal	No	Si
Se ven las tarifas del personal a contratar	No	Si
Módulo de Finca		
Dispone de detalles específicos, como ubicación, tamaño, o características	Si	Si
Dispone de imágenes de la finca	Si	Si
Módulo de Contacto		
Se proporciona un formulario de contacto	Si	Si
Módulo de Login		
Se dispone de registro y login en la plataforma	No	Si
Módulo de Carrito		
Dispone de control de unidades de productos	No	Si
Dispone de opciones de pago	Si	Si
Módulo de Reservas		
Se dispone de un histórico de las reservas realizadas y su estatus	No	Si
Módulo de Perfil		
El usuario puede gestionar su perfil personal	No	Si

2.3.- VALORACIÓN

En el apartado anterior se han revisado tres aplicaciones, sin embargo, ninguna de ellas cumple satisfactoriamente con todos los objetivos y funcionalidades deseados. A nivel de diseño y usabilidad, todas presentan ciertos puntos fuertes, aunque "Ya" podría mejorar su interfaz y ser más intuitiva y fácil de navegar.

Al analizar las limitaciones de cada aplicación, se observa que tanto "La Seu" como "Ya" tienen deficiencias en aspectos cruciales como la categorización de productos, control de unidades y precios, gestión de personal, y la falta de un sistema de registro y login, lo que repercute negativamente en la experiencia del usuario al no poder acceder a un historial de reservas.

Por otro lado, "CaterSpot" no permite la selección de personal ni muestra tarifas correspondientes, al igual ocurre con esta característica en "Ya". Además, "CaterSpot" es la única que no cuenta con la opción de búsqueda de fincas o espacios para eventos. "La Seu" y "Ya" sí cuentan con esta opción, aunque carecen de detalles específicos sobre las fincas.

Por todo ello, "Chapata" sobresale al ofrecer una experiencia más completa y funcional para el usuario. Especialmente en lo que respecta a la categorización de productos, contratación de personal, registro y login,

En conclusión, ninguna de las aplicaciones analizadas cumple con todos los requisitos deseados, pero "Chapata" se destaca como la opción más completa y funcional hasta el momento pese a falta de desarrollar una pasarela de pago.

Capítulo 3

Hipótesis de trabajo



3.1.- HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo son programas o aplicaciones que se utilizan en el desarrollo de un software. En este proyecto se han utilizado las siguientes:

3.1.1.- Spring Tool Suite 3

La herramienta Spring Tool Suite (STS) nace con la idea de facilitar la configuración de un entorno de desarrollo ya que a día de hoy hay tantas opciones y configuraciones posibles que puede ser una tarea abrumadora, especialmente para aquellos que son nuevos en el mundo de TI. Spring Tool Suite (STS) es un entorno de desarrollo integrado (IDE) basado en Eclipse, diseñado específicamente para implementar, ejecutar, depurar y desplegar las aplicaciones utilizando el framework Spring Boot

La herramienta incluye el servidor Pivotal tc Server, una versión de Apache Tomcat optimizado para Spring. Con su consola Spring Insight, tc Server Developer Edition, desde la cual ofrece una visión gráfica en tiempo real de los parámetros de rendimiento de las aplicaciones lo que permite a los desarrolladores identificar y diagnosticar los problemas desde sus escritorios.[10]



Figura 3.1:Logo Spring Tool Suite

Ofrece una variedad de de características que pueden ayudar a mejorar la productividad del desarrollador, tales como:

- Plantillas de proyecto predefinidas para crear nuevos proyectos con Spring Boot, MVC y Batch, lo que acelera el proceso de inicio de un nuevo proyecto.
- Generación automática de metadatos facilitando la creación de archivos de configuración del proyecto, como el archivo pom.xml en el caso de proyectos Maven.
- Soporte para la recarga dinámica integrando una función para la recarga de clases y recursos de Java de forma dinámica, lo que permite ver los cambios realizados en el código de forma instantánea sin necesidad de reiniciar la aplicación.
- Integración con herramientas de construcción populares como pueden ser Maven y Gradle, lo que simplifica la gestión de dependencias y la construcción del proyecto.
- Vista de consola desde la que se puede ver la salida de los registros de la aplicación, facilitando la depuración y el monitoreo del comportamiento de la aplicación en tiempo de ejecución. [11]

Además, Spring Tool Suite proporciona una gran cantidad de complementos que pueden agregar aún más funciones, como:

- Spring Boot: Ofrece una visión orientada a la creación de aplicaciones Spring ofreciendo un funcionamiento lo más rápido posible.
- Spring Framework: Proporciona soporte central para la inyección de dependencias, gestión de transacciones, aplicaciones web, acceso a datos, mensajería, entre otros.
- Spring Data: Ofrece un enfoque consistente para el acceso a datos, ya sea de un origen relacional, no relacional, o de otros tipos.

- Spring Cloud: Proporciona un conjunto de herramientas para patrones comunes en sistemas distribuidos, útil para construir y desplegar microservicios.
- Spring Cloud Data Flow: Ofrece un servicio de herramientas para aplicaciones de microservicios de datos componibles en entornos modernos.
- Spring Security: Protege tu aplicación con un soporte integral y extensible para autenticación y autorización.
- Spring Authorization Server: Proporciona una base segura, ligera y personalizable para construir proveedores de identidad OpenID Connect 1.0 y servidores de autorización OAuth2.
- Spring for GraphQL: Ofrece soporte para aplicaciones Spring construidas en GraphQL Java.
- Spring Session: Proporciona una API e implementaciones para gestionar la información de la sesión del usuario.
- Spring Integration: Admite los conocidos Patrones de Integración Empresarial a través de mensajería ligera y adaptadores declarativos.[12]

Por último, añadir que Spring Tool Suite es de libre acceso para el desarrollo y sin límite de tiempo, con código completamente abierto y bajo los términos de la Licencia Pública Eclipse. Se puede descargar desde su página web <https://spring.io/tools> donde hay dos versiones disponibles:

1. Edición estándar la cual incluye las funciones principales de STS.
2. Edición ultimate que incluye todas las funciones de la edición estándar, además de complementos adicionales para desarrollar aplicaciones nativas de la nube.

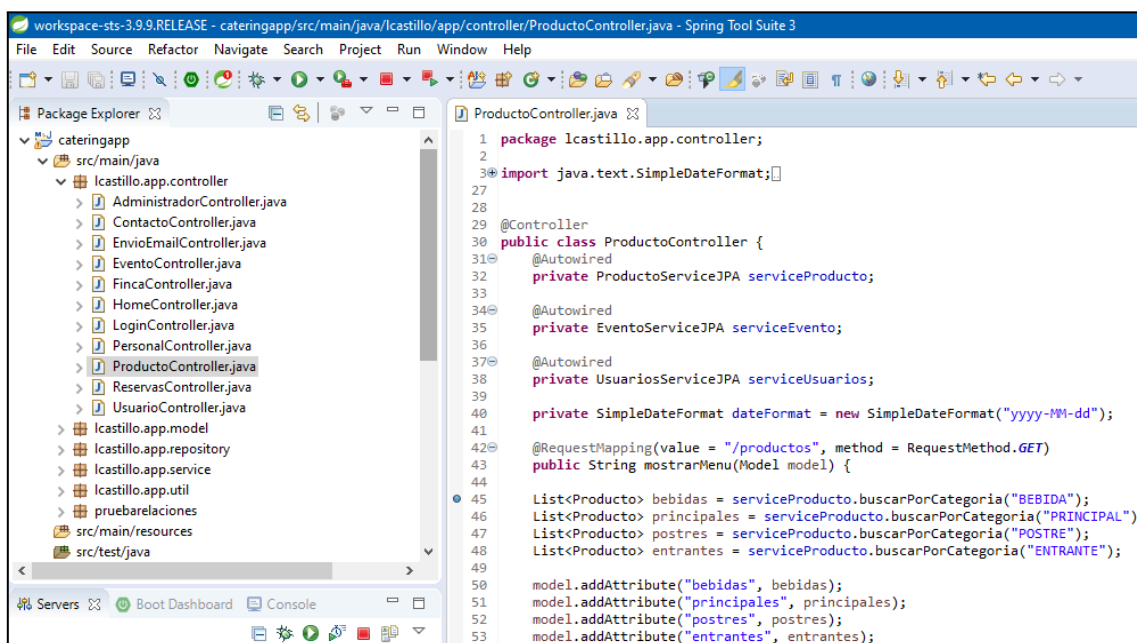


Figura 3.2: Spring Tool Suite 3 (versión 3.9.9.)

3.1.2.- Arquitectura WAMP

Esta arquitectura, se podría definir del siguiente modo: *“WAMP es un acrónimo que significa Windows, Apache, MySQL y PHP. Es un stack o conjunto de soluciones de software que cuando instalas WAMP, con él se instala Apache, MySQL y PHP en el sistema operativo”* [13] (en mi caso es Windows de ahí que se trate de una arquitectura WAMP). Cabe destacar que el concepto WAMP deriva de LAMP (la “L” significa Linux). La única diferencia entre ambos es que WAMP se usa para Windows mientras que LAMP para sistemas operativos basados en Linux. El significado de cada letra del acrónimo es el siguiente:

- “W” se refiere al sistema operativo Windows (también existen los términos ‘LAMP’ para Linux y ‘MAMP’ para MAC).
- “A” es la inicial de Apache. Apache es el software que actúa como servidor y que se encarga de servir las páginas web solicitadas. Cuando se realiza una petición a una página web, Apache cumple la solicitud utilizando el protocolo HTTP y muestra el sitio correspondiente.
- “M” significaba inicialmente MySQL, aunque en la actualidad también puede referirse a MariaDB. Tanto MySQL como MariaDB son sistemas de gestión de base de datos para el lado del servidor. Almacenan toda la información relevante, contenidos del sitio, perfiles de usuarios, etc.
- “P” significa PHP. Se trata de un lenguaje de script del lado del servidor, por tanto, se ejecuta junto con Apache y se comunica con MySQL mediante una librería nativa. Actualmente, dicha librería también sirve para comunicarse con MariaDB.

3.1.2.1.- XAMPP

Se trata de un servidor independiente de plataforma de código abierto. Permite instalar fácilmente Apache en el propio ordenador, sin importar el sistema operativo, ya que se utilizan versiones para varios de ellos (Linux, Windows, MAC o Solaris) [14]. Incluye el servidor de base de datos MariaDB (en versiones anteriores, MySQL) y gestor phpMyAdmin. También incluye, entre otras opciones, intérpretes para los lenguajes PHP y Perl, y el servidor de FTP FileZilla.

XAMPP permite, con mucha facilidad, crear un servidor localhost que posibilita desarrollar y probar páginas web en la propia máquina sin necesidad de tener que acceder a Internet y tener un dominio.

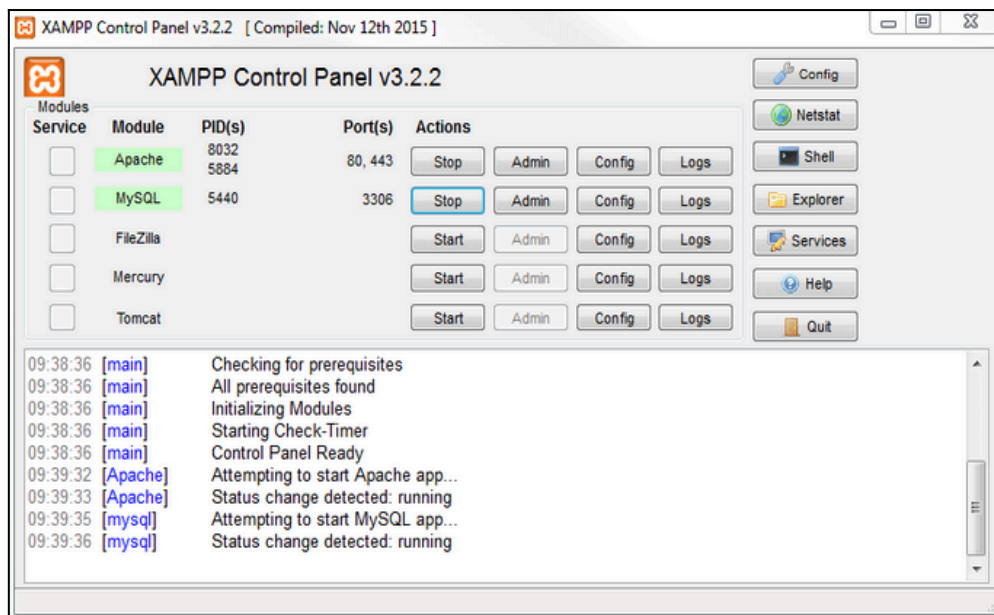


Figura 3.3: Panel de control XAMPP

3.1.3.- phpMyAdmin

phpMyAdmin es una herramienta de software gratuita desarrollada en PHP. Originalmente se ideó para la administración de MySQL vía web. En la actualidad, permite una gran gama de operaciones sobre MySQL, pero también sobre MariaDB.[\[16\]](#)

Con phpMyAdmin se puede realizar todo tipo de operaciones de una manera visual y muy intuitiva, desde la administración de las tablas (crear, modificar y eliminar), como para manipular datos (crear, borrar, modificar, ejecutar consultas).

El acceso a phpMyAdmin es vía web, esto quiere decir que se aloja en el propio servidor y se puede acceder desde cualquier dispositivo con conexión a Internet en lugar de usar un único ordenador. Algunas de sus características más importantes son:

- Interfaz web intuitiva
- Soporte para la mayoría de las características de MySQL y MariaDB
- Importa y exporta datos de CSV y SQL
- Administrar múltiples servidores
- Crea gráficos de diseño de la base de datos en varios formatos
- Busca de forma global en una base de datos o en un subconjunto
- Transforma los datos almacenados en cualquier formato utilizando un conjunto de funciones predefinidas.

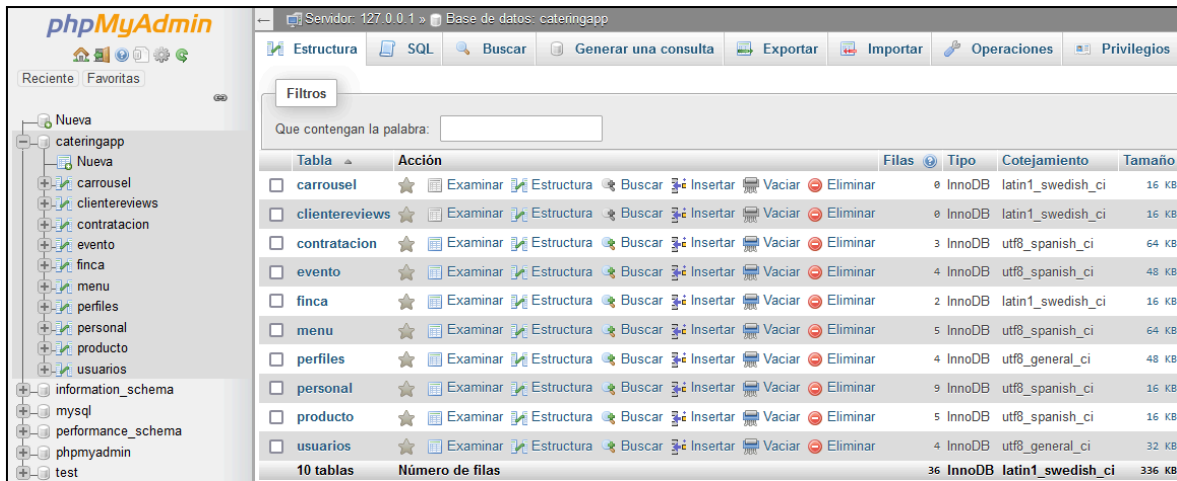


Figura 3.4: Panel de control phpMyAdmin

3.1.4.- Dia

Dia es una aplicación informática cuyo propósito general es la creación de diagramas. Inicialmente fue desarrollada por Alexander Larsson y continuada como parte del proyecto GNOME. Está pensada como un sustituto de la aplicación comercial Visio de Microsoft. Su estructura es modular por lo que ofrece diferentes paquetes de formas adaptados a las distintas necesidades de diseño.

Se puede emplear para crear diferentes tipos de diagramas, como diagramas entidad-relación, diagramas UML, diagramas de flujo, de diagramas redes, y de circuitos eléctricos, entre otros. Además, es sencillo agregar nuevas formas: simplemente dibujandolas utilizando un subconjunto de SVG e integrarlas al archivo XML. Con la ayuda del paquete dia2code, es posible generar el esqueleto del código a escribir, si se utiliza con ese propósito, creando diagramas UML.[17]

El formato para leer y almacenar gráficos es XML (comprimido con gzip, para ahorrar espacio). Puede producir salida en los formatos EPS, SVG y PNG. Dia puede exportar diagramas a varios formatos incluyendo los siguientes:

- EPS (Encapsulated PostScript)
- SVG (Scalable Vector Graphics)
- DXF (Autocad's Drawing Interchange Format)
- CGM (Computer Graphics Metafile definido por estándares ISO)
- WMF (Windows Meta File)
- PNG (Portable Network Graphics)
- JPEG (Joint Photographic Experts Group)
- VDX (Microsoft's XML para Visio Drawing)

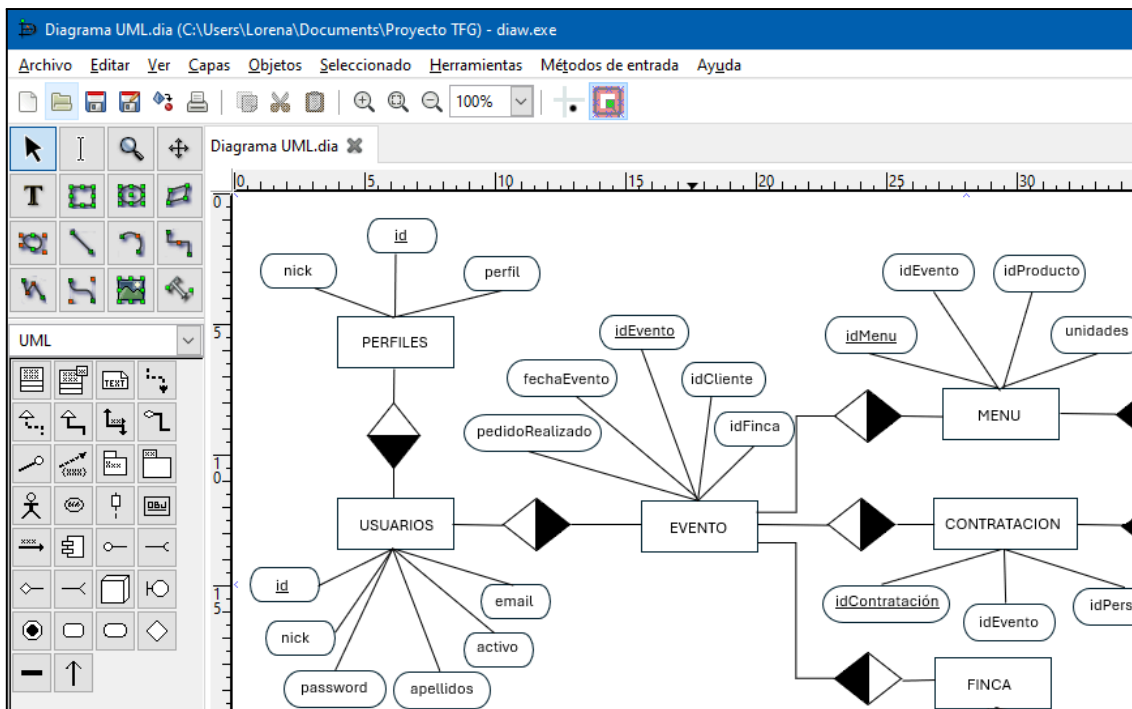


Figura 3.5: Interfaz de la herramienta Dia

En este proyecto se ha utilizado para la generación del diagrama Entidad/Relación y para el diagrama de clases.

3.1.5.- GanttProject

GanttProject es un software de código abierto bajo licencia GPL desarrollado en Java con la biblioteca Swing. El proyecto GanttProject inició en enero de 2003 en la University de Marne-la-Vallée, Francia, bajo la dirección inicial de Alexandre Thomas y posteriormente bajo la dirección de Dmitry Barashev. El software es compatible con sistemas operativos como Windows, Linux y Mac OS X, lo que lo hace accesible para una amplia variedad de usuarios.[18]

Su propósito principal es la gestión de proyectos mediante el uso del diagrama de Gantt. Este tipo de diagrama permite programar tareas de manera consecutiva o simultánea y facilita la administración de recursos a través de diagramas de carga de recursos.

Actualmente, GanttProject está diseñado para manejar unidades de tiempo a nivel de días, no de horas. Permite generar diversos tipos de informes compatibles con herramientas como MS Project, así como exportar datos a formatos como HTML, PDF y hojas de cálculo.[19]

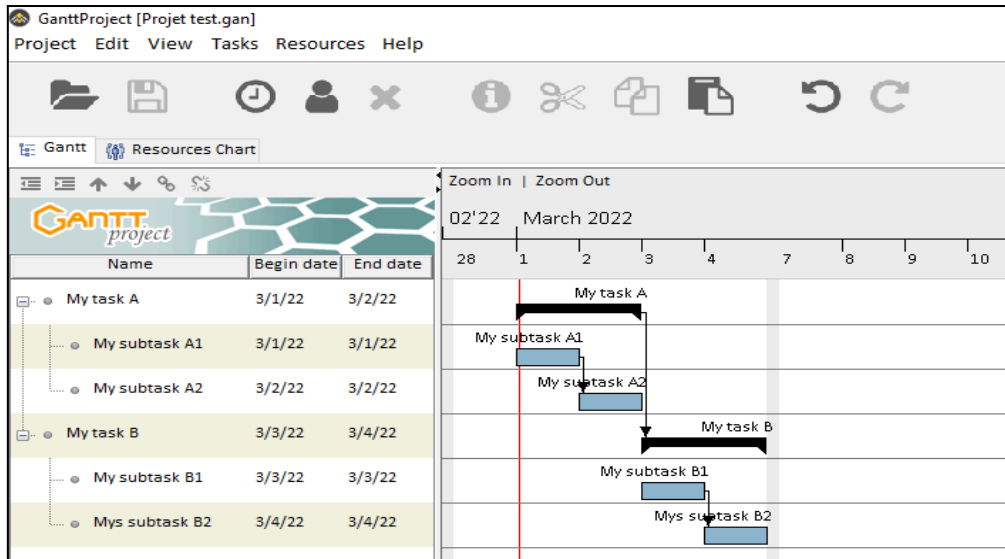


Figura 3.6: Interfaz de la GanttProject

3.1.6.- Maven Project

Maven fue creada por Jason van Zyl, en 2002, es una herramienta de software para la gestión y construcción de proyectos completos desde la etapa en la que se comprueba que el código es correcto, hasta que se despliega la aplicación, pasando por la ejecución de pruebas y generación de informes y documentación.

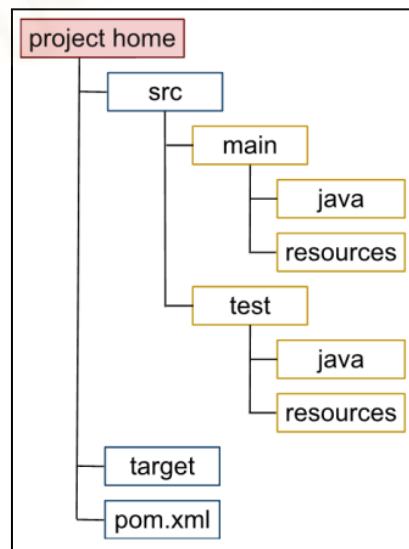


Figura 3.7: Estructura de un proyecto con Maven [21]

Aunque se puede utilizar con diversos lenguajes (C#, Ruby, Scala...) se usa principalmente en proyectos Java, donde es muy común ya que está escrita en este lenguaje. De hecho, frameworks Java como Spring y Spring Boot la utilizan por defecto.

Es similar en cuanto a funcionalidad a Apache Ant (y en menor medida a PEAR de PHP y CPAN de Perl), pero tiene un modelo de configuración de construcción más simple, utilizando un Project Object Model (POM) para describir el proyecto de software a construir, lo que le hace ser una herramienta declarativa [20]. Es decir, todo lo que especifiquemos, incluyendo las dependencias en módulos y componentes externos, los procesos de compilación, el orden en que se ejecutan, así como los plugins específicos de Maven que utilizemos se almacenan en el archivo pom.xml. Este archivo es el que Maven lee a la hora de funcionar. Actualmente es proyecto de nivel superior de la Apache Software Foundation. La figura 3.8 muestra un trozo del contenido del archivo pom.xml de este proyecto.



```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4   http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <groupId>lcastillo</groupId>
7   <artifactId>cateringapp</artifactId>
8   <version>0.0.1-SNAPSHOT</version>
9
10  <dependencies>
11    <dependency>
12      <groupId>org.springframework</groupId>
13      <artifactId>spring-webmvc</artifactId>
14      <version>5.2.1.RELEASE</version>
15    </dependency>
16
17    <dependency>
18      <groupId>org.springframework.data</groupId>
19      <artifactId>spring-data-jpa</artifactId>
20      <version>2.2.0.RELEASE</version>
21    </dependency>
22
23    <dependency>
24      <groupId>mysql</groupId>
25      <artifactId>mysql-connector-java</artifactId>
26      <version>5.1.48</version>
27    </dependency>
28
29    <dependency>
30      <groupId>org.webjars</groupId>
31      <artifactId>bootstrap</artifactId>
32      <version>4.6.0</version>
33    </dependency>
```

Figura 3.8: Archivo pom.xml

Los 5 primeros nodos del XML (figura 3.8) son los únicos obligatorios y definen respectivamente el modelo de objetos que utilizará Maven:

1. <project>: Esta etiqueta raíz define el inicio del archivo POM. Contiene información general sobre el proyecto.
2. <modelVersion>: Indica la versión del modelo POM. En este caso, se utiliza la versión 4.0.0.

3. `<groupId>`: Identifica el grupo o la organización que es responsable del proyecto. En tu caso, el grupo es lcastillo.
4. `<artifactId>`: Es el nombre único del proyecto. Aquí, se llama cateringapp.
5. `<version>`: Es la versión actual del proyecto. La versión es 0.0.1-SNAPSHOT.

Como se puede observar incluye una serie de dependencias que especifican otras bibliotecas que este proyecto necesita para compilar, ejecutar o funcionar correctamente. Están representadas por la etiqueta `<dependency>`. Las dependencias mostradas son:

- `spring-webmvc`: Es parte del framework Spring para aplicaciones web.
- `spring-data-jpa`: Facilita la interacción con bases de datos mediante JPA (Java Persistence API).
- `mysql-connector-java`: Conector JDBC para MySQL.
- `bootstrap`: Versión 4.6.0. Framework CSS para diseño web.

3.2.- FRAMEWORK Y LIBRERÍAS

Llamamos librería a un conjunto de funciones, clases y métodos, escritos en un lenguaje de programación específico, que presentan una interfaz o un contrato definido para una funcionalidad en particular. El propósito principal de una librería es ejecutar una acción concreta para reutilizar el código, lo que facilita su uso por parte de otras aplicaciones o librerías. Por otro lado, un framework (del inglés, “marco de trabajo”) es un concepto superior al de librería ya que incluye un conjunto de librerías y subconjuntos de framework que aceleran el desarrollo de una aplicación. Algunos frameworks incluso proporcionan una pila completa de herramientas, lo que significa que no se necesita instalar librerías adicionales para construir una aplicación. Otros son microframeworks, que ofrecen solo funcionalidades básicas y permiten agregar plugins o librerías adicionales según ciertas necesidades específicas.[\[22\]](#)

Globalmente, ambos elementos (librerías y frameworks) buscan poder escribir código o desarrollar una aplicación de manera más sencilla, rápida y eficiente. Es algo que permite una mejor organización y control de todo el código elaborado, garantizando una mayor productividad que los métodos más convencionales y una minimización del coste al agilizar las horas de trabajo volcadas en el desarrollo.[\[23\]](#)

3.2.1.- Spring boot

Antes de adentrarnos en Spring Boot, es necesario hablar sobre Spring Framework, ya que existe una relación muy cercana entre ambos.

Spring Framework proporciona un soporte de infraestructura completo para el desarrollo de aplicaciones empresariales en Java. Tiene características muy buenas como inyección de dependencias y módulos que reducen el tiempo de desarrollo de una aplicación [24]. Los módulos principales de que dispone Spring Framework actualmente son:

- Data Access/Integration (Acceso a Datos/Integración): Este módulo proporciona un conjunto de abstracciones y clases para trabajar con bases de datos y realizar operaciones de acceso a datos de manera más sencilla y consistente.
- Web: Ofrece soporte para el desarrollo de aplicaciones web, incluyendo la gestión de peticiones HTTP, la creación de controladores y la integración con tecnologías web como Servlets y JSP.

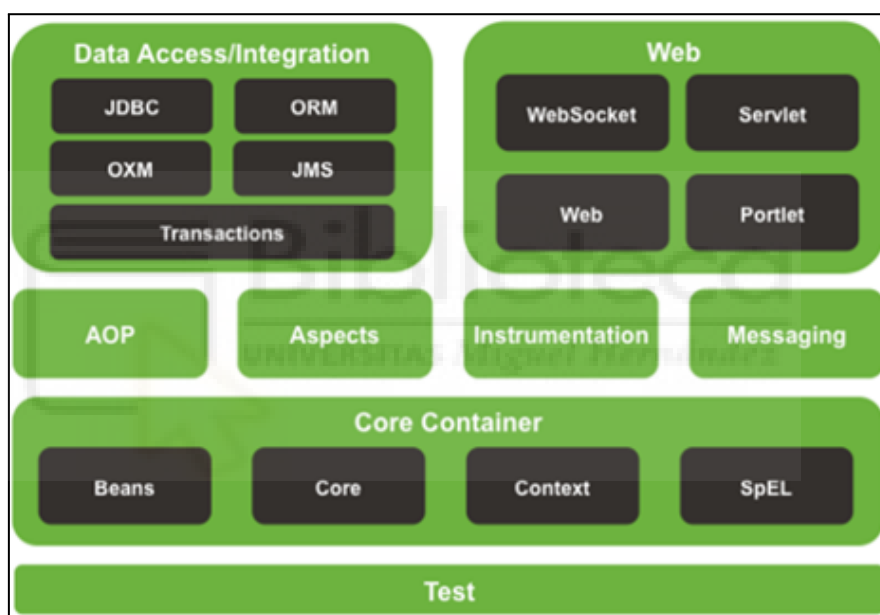


Figura 3.9: Módulos Spring

- AOP (Aspect-Oriented Programming - Programación Orientada a Aspectos): Permite separar preocupaciones transversales (como la seguridad o la gestión de transacciones) del código de la aplicación principal mediante la definición de aspectos que se aplican de manera transparente en varios puntos de la aplicación.
- Aspect: Este módulo proporciona una implementación específica de AOP que se integra con el contenedor de Spring para la gestión de aspectos.
- Instrumentation (Instrumentación): Ofrece herramientas para instrumentar y supervisar aplicaciones en tiempo de ejecución, lo que permite recopilar métricas y datos de rendimiento.

- Messaging (Mensajería): Proporciona soporte para la integración de sistemas a través de mensajes, incluyendo la implementación de patrones de mensajería como publish-subscribe y message queues.
- Core Container (Módulo Principal): Contiene los componentes fundamentales de Spring Framework, incluyendo el contenedor de inversión de control (IoC) y el contenedor de beans, que gestionan la creación y la administración de los objetos de la aplicación.
- Test: Ofrece herramientas y clases de utilidad para escribir y ejecutar pruebas unitarias e integración en aplicaciones basadas en Spring.

Cada uno de estos módulos proporciona funcionalidades específicas que se pueden utilizar individualmente o en conjunto para desarrollar una amplia gama de aplicaciones.

Spring Boot, por otro lado, es una extensión del Spring Framework que facilita el proceso de configuración y desarrollo de aplicaciones Spring. Proporciona un conjunto de dependencias preconfiguradas y una estructura de proyecto convencional para desarrollar aplicaciones de forma rápida y sencilla. Spring Boot también incluye un servidor integrado (por lo general, Tomcat) para ejecutar aplicaciones de manera independiente, lo que significa que no es necesario configurar un servidor externo. De esta forma queda eliminada gran parte de la configuración manual que sí sería necesaria en el Spring Framework tradicional.[\[25\]](#)

Para la estructura y configuración de este proyecto, contamos con dos archivos. El primer archivo es el "pom.xml", que es donde se gestionan las dependencias del proyecto. Este archivo es crucial ya que permite definir y gestionar las bibliotecas y dependencias que necesita la aplicación para funcionar correctamente. Aquí especificamos las versiones de las bibliotecas de Spring, Hibernate, MySQL, y otras dependencias necesarias para el proyecto.

El segundo archivo es el "root-context.xml" (figura 3.10), donde se definen los beans de la aplicación. En la terminología de Spring, los objetos gestionados por el contenedor de Spring se conocen como beans. Este archivo es fundamental en la configuración de la aplicación ya que establece varios componentes esenciales que se describirán a continuación.

En primer lugar, incluye el escaneo de componentes a través de la etiqueta `<context:component-scan>`. Este escaneo es importante porque permite que Spring detecte automáticamente los componentes marcados con anotaciones específicas, como `@Service`, `@Repository`, `@Controller`, entre otros.

A continuación se especifica la configuración del DataSource que es un componente clave que define la información de conexión a la base de datos, como la URL de la base de datos, el nombre de usuario y la contraseña.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:context="http://www.springframework.org/schema/context"
5       xmlns:jpa="http://www.springframework.org/schema/data/jpa"
6       xmlns:tx="http://www.springframework.org/schema/tx"
7       xsi:schemaLocation="http://www.springframework.org/schema/beans
8         http://www.springframework.org/schema/beans/spring-beans.xsd
9         http://www.springframework.org/schema/context
10        http://www.springframework.org/schema/context/spring-context-4.3.xsd
11        http://www.springframework.org/schema/data/jpa
12        http://www.springframework.org/schema/data/jpa/spring-jpa-1.8.xsd
13        http://www.springframework.org/schema/tx
14        http://www.springframework.org/schema/tx/spring-tx-4.3.xsd">
15
16     <context:component-scan base-package="lcastillo.app.service"></context:component-scan>
17     <jpa:repositories base-package="lcastillo.app.repository" />
18
19     <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
20       <property name="driverClassName" value="com.mysql.jdbc.Driver" />
21       <property name="url" value="jdbc:mysql://localhost:3306/cateringapp?useSSL=false" />
22       <property name="username" value="root" />
23       <property name="password" value="" />
24     </bean>
25     <bean id="jpaVendorAdapter" class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
26       <property name="generateDdl" value="false" />
27       <property name="showSql" value="true"></property>
28       <property name="databasePlatform" value="org.hibernate.dialect.MySQL5Dialect" />
29     </bean>
30     <bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
31       <property name="packagesToScan" value="lcastillo.app.model" />
32       <property name="dataSource" ref="dataSource" />
33       <property name="jpaVendorAdapter" ref="jpaVendorAdapter" />
34     </bean>
```

Figura 3.10: Archivo root-context.xml

Para finalizar la breve descripción de este archivo, haremos referencia a uno de los componentes más destacados que es la configuración de repositorios JPA. Los repositorios JPA son interfaces para interactuar con las entidades de la base de datos de manera más fácil y eficiente. En este contexto, Spring Data JPA proporciona la capacidad de definir estos repositorios mediante interfaces, y luego Spring se encarga de generar automáticamente la implementación necesaria en tiempo de ejecución. Esto simplifica enormemente el acceso a los datos y permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de manera más sencilla.[\[26\]](#)

Esta combinación mixta de archivos XML y configuración basada en anotaciones permite un control completo sobre la configuración y las dependencias de la aplicación, asegurando su correcto funcionamiento y mantenimiento (ver figura 3.11). La anotación `@Controller` se utiliza para marcar una clase como un controlador en Spring MVC (Modelo-Vista-Controlador). Esto indica que la clase se encargará de manejar las solicitudes HTTP entrantes y devolver las respuestas apropiadas. Los métodos dentro de una clase anotada con `@Controller` pueden ser mapeados a las URL de solicitud utilizando otras anotaciones como `@RequestMapping`, `@GetMapping`, `@PostMapping`, etc. Los controladores en Spring MVC son responsables de interactuar con los datos del modelo, procesar las solicitudes del usuario y devolver las vistas o los datos JSON apropiados.

```
PersonalController.java
1 package lcastillo.app.controller;
2 import java.text.SimpleDateFormat;
25
26 @Controller
27 public class PersonalController {
28     @Autowired
29     private PersonalServiceJPA servicePersonal;
30
31     @Autowired
32     private EventoServiceJPA serviceEvento;
33
34     @Autowired
35     private UsuariosServiceJPA serviceUsuarios;
36
37     @RequestMapping(value = "/personal", method = RequestMethod.GET)
38     public String mostrarPersonal(Model model) {
39         List<Personal> empleados = servicePersonal.buscarTodas();
40         model.addAttribute("empleados", empleados);
41
42         return "personal";
43     }
```

Figura 3.11: Archivo PersonalControler.java

La anotación `@Autowired` se utiliza para realizar la inyección de dependencias en Spring. Permite que Spring automáticamente resuelva y suministre las dependencias de un bean en tiempo de ejecución. Cuando se coloca la anotación `@Autowired` sobre un campo, método setter o constructor en una clase, Spring buscará una instancia compatible del tipo requerido y la inyectará automáticamente en ese punto.

En este caso, se muestra cómo se inyectan automáticamente las instancias de los servicios `PersonalServiceJPA`, `EventoServiceJPA` y `UsuariosServiceJPA` en el controlador. Estos servicios proporcionan funcionalidades específicas relacionadas con la gestión de datos en la aplicación.[\[27\]](#)

3.2.2.- Bootstrap

Bootstrap es un popular framework de código abierto para el desarrollo front-end de sitios web y aplicaciones web. Surgió en 2010 bajo el nombre de Twitter Blueprint, ya que fue creado para desarrollar Twitter. Sin embargo, tras su lanzamiento público en 2011, cambió su nombre por el de Bootstrap. Actualmente este framework ofrece una colección de herramientas y componentes HTML, CSS y JavaScript que facilitan la creación de interfaces de usuario atractivas, responsivas y consistentes de manera rápida y sencilla. Entre las características principales de Bootstrap se incluyen [\[28\]](#):

- Grid System: Proporciona un sistema de rejilla flexible y fácil de usar que permite crear diseños responsivos y adaptables a diferentes tamaños de pantalla, desde dispositivos móviles hasta ordenadores de escritorio.
- Componentes Reutilizables: Ofrece una amplia gama de componentes predefinidos, como botones, formularios, navegación, barras de progreso, tarjetas, entre otros, que pueden ser fácilmente integrados en cualquier proyecto web.
- Estilos y Temas: Incluye un conjunto de estilos CSS predefinidos que ayudan a mejorar la apariencia visual de las aplicaciones web. Además, ofrece la posibilidad de personalizar y crear temas personalizados a través de variables CSS.
- JavaScript Plugins: Cuenta con una colección de plugins JavaScript que añaden diversas funcionalidades interactivas y dinámicas a los componentes.
- Compatibilidad con Navegadores: Está diseñado para funcionar de manera consistente en una amplia gama de navegadores web modernos, lo que garantiza una experiencia de usuario coherente y confiable.

Con Bootstrap se consigue ahorrar mucho tiempo en el proceso de diseño y maquetación de una página web. Debido a su funcionamiento ya no hace falta crear clases y adaptarlas en un archivo .css, sino que las propias clases que proporciona Bootstrap, una vez asignadas a los elementos HTML, ya incorporan los estilos apropiados para obtener la maquetación deseada.

3.2.3.- Librería Java Standard Library

La Java Standard Library, también conocida como Java Standard Edition (Java SE) API, es un conjunto de clases y métodos predefinidos por los creadores de Java para facilitar el desarrollo de aplicaciones en este lenguaje. Estas clases proporcionan funcionalidades básicas y esenciales para el desarrollo de aplicaciones siendo estos los fundamentos necesarios para construir aplicaciones robustas y eficientes en la plataforma Java. La Java Standard Library abarca una amplia variedad de áreas clave en el desarrollo de aplicaciones Java. Estos son algunos aspectos más destacados sobre esta biblioteca [29]:

- java.lang: Contiene clases básicas que se importan automáticamente (implícitamente) en todas las aplicaciones Java. Algunos ejemplos son String, Integer, Double, etc.
- java.util: Incluye clases para estructuras de datos como listas, mapas, conjuntos, así como utilidades para manejar fechas, horas, entre otras opciones.

- `java.io`: Proporciona clases para operaciones de entrada/salida, como lectura y escritura de archivos.
- `java.util.concurrent`: Proporciona clases para programación concurrente y manejo de hilos.
- `java.security`: Incluye clases para seguridad, como cifrado, autenticación y manejo de claves.
- `java.sql`: Ofrece clases para interactuar con bases de datos mediante JDBC (Java Database Connectivity).
- `java.time`: Introduce nuevas clases para manejar fechas, horas y zonas horarias (disponible a partir de Java 8).
- `javax.xml`: Contiene clases para procesar y manipular datos XML.

3.2.4.- Librería **JavaServer Pages Standard Tag Library**

La **JavaServer Pages Standard Tag Library (JSTL)** es una colección de etiquetas personalizadas (tags) que proporciona funcionalidades comunes y reutilizables para el desarrollo de aplicaciones **JavaServer Pages (JSP)**. JSTL proporciona cuatro bibliotecas de etiquetas (Tag Libraries) con utilidades ampliamente utilizadas en el desarrollo de páginas web dinámicas [30]:

1. **Core Tags (<c:>)**: Estas etiquetas simplifican tareas comunes como la iteración sobre listas, condiciones y manipulación de variables. Algunas de las etiquetas más comunes incluyen `forEach` para iterar sobre colecciones, `if` para la ejecución condicional, y `set` para la asignación de valores a variables.
2. **XML Tags (<x:>)**: Proporciona etiquetas que facilitan el procesamiento de documentos XML en JSP. Pueden ser utilizadas para la lectura, escritura y transformación de documentos XML.
3. **Formatting Tags (<fmt:>)**: Ofrece etiquetas para el formateo y presentación de datos, incluyendo fechas, números y monedas, de acuerdo a diferentes locales y preferencias de formato.
4. **SQL Tags (<sql:>)**: Permite ejecutar consultas SQL directamente en páginas JSP. Aunque no se recomienda su uso directo debido a consideraciones de seguridad y separación de capas, estas etiquetas pueden ser útiles en situaciones específicas donde se necesita una integración directa con la base de datos desde una página JSP.

Estas etiquetas simplifican la creación de páginas JSP al encapsular la lógica de programación repetitiva, de esta manera se consigue reducir la necesidad de código Java en las páginas JSP, con lo que se logra mejorar la legibilidad y mantenibilidad del código desarrollado.

3.2.5.- Librería JavaMail API

La librería JavaMail API se utiliza para crear aplicaciones que envían, reciben y gestionan correos electrónicos. Proporciona una interfaz para trabajar con protocolos de correo electrónico como SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3) e IMAP (Internet Message Access Protocol). Además de enviar correos electrónicos, la JavaMail API también permite a las aplicaciones Java leer mensajes de correo electrónico de servidores de correo remotos, lo que puede ser útil para desarrollar clientes de correo electrónico personalizados o integrar funcionalidades de correo electrónico en aplicaciones empresariales.[31]

3.2.6.- Librería MySQL Connector/J

El MySQL Connector/J es una librería y a su vez es un controlador. Es el controlador oficial JDBC (Java Database Connectivity) que permite a las aplicaciones Java comunicarse con bases de datos MySQL utilizando el estándar JDBC. Este conector es esencialmente un puente entre una aplicación Java y un servidor MySQL, facilitando la ejecución de consultas, actualizaciones y otras operaciones de base de datos desde el código Java. Esto proporciona una interfaz de programación simple y familiar para realizar operaciones de base de datos de manera eficiente y segura.[32]

3.3.- ARQUITECTURA MVC

El patrón Modelo-Vista-Controlador (MVC) es una arquitectura de diseño de software utilizada para separar la lógica la aplicación en tres componentes principales [33]:

- **Modelo (Model):** El Modelo representa los datos y la lógica de negocio de la aplicación y suele ser representado por clases Java. Es responsable de la manipulación de datos, procesar la información y proporcionarla al controlador.
- **Vista (View):** La Vista es la interfaz de usuario con la que interactúan los usuarios. Presenta los datos al usuario de una manera legible y comprensible. En el desarrollo web, la vista suele ser una página HTML, una plantilla como Thymeleaf o JSP para crear la interfaz de usuario.
- **Controlador (Controller):** El Controlador actúa como intermediario entre el modelo y la vista. Recibe las solicitudes HTTP del cliente, ejecuta la lógica de la aplicación y decide qué vista mostrar al usuario.

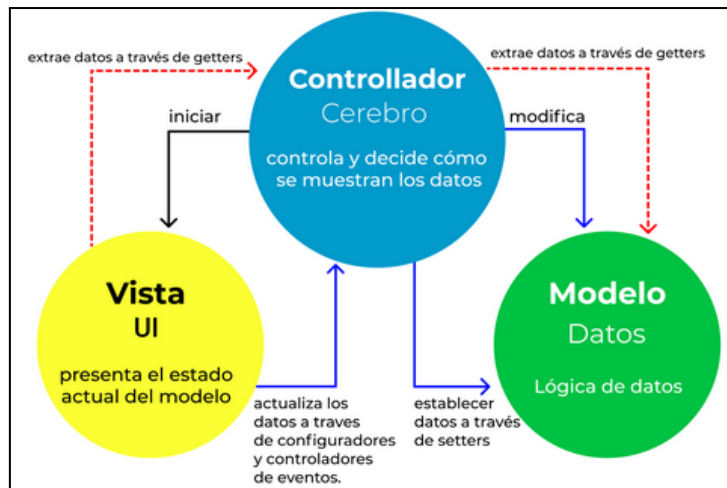


Figura 3.12: Patrón MVC

En el contexto de Spring Boot, el patrón MVC se implanta de la siguiente manera:

```

// Clase de negocio (Modelo)
package lcastillo.app.modelo;

public class Producto {
    private String nombre;
    private String categoria;
    private Float precio;
    // Getters y setters (omito para simplificar)
}

// Controlador
package lcastillo.app.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import lcastillo.app.modelo.Producto;
import lcastillo.app.service.ProductoServiceJPA;

@Controller
public class ProductoController {
    @Autowired
    private ProductoServiceJPA serviceProducto;

    @RequestMapping(value = "/productos", method = RequestMethod.GET)
    public String mostrarMenu(Model model) {

        // Carga de datos (en memoria)
        List<Producto> bebidas = serviceProducto.buscarPorCategoria("BEBIDA");
        List<Producto> principales = serviceProducto.buscarPorCategoria("PRINCIPAL");

        // Agregar los productos al modelo
        model.addAttribute("bebidas", bebidas);
        model.addAttribute("principales", principales);

        // Devolver la vista JSP
        return "productos"; // Nombre de la plantilla
    }
}
  
```

Figura 3.13: Ejemplo implantación MVC con Spring Boot

En este ejemplo el controlador `ProductoController` maneja la solicitud de la URL `/productos`. Crea los objetos `Productos` y los agrega al modelo. Devuelve la vista llamada "productos". Así es como el controlador comunica datos entre el modelo y la vista utilizando el objeto `Model` y como la vista puede acceder a los datos proporcionados por el controlador y mostrarlos al usuario.

3.4.- TECNOLOGÍA FRONTEND

Las tecnologías frontend son el conjunto de herramientas y técnicas utilizadas para crear la estructura, el diseño y la interactividad de un sitio web o aplicación. Estas tecnologías se centran en lo que los usuarios ven y con lo que interactúan directamente en un sitio web. En este proyecto se han utilizado las siguientes tecnologías frontend:

- **HTML:** HTML5 (HyperText Markup Language, versión 5) es la quinta versión más importante del lenguaje básico de la World Wide Web, HTML. Su objetivo principal es definir la estructura y el contenido de una página web. No debemos confundir HTML5 con un lenguaje de programación ya que, en realidad, es un lenguaje de marcado que estructura, interpreta y despliega el contenido de un documento web mediante etiquetas y atributos (no permite definir flujos de control como los lenguajes de programación convencionales). A diferencia de las versiones anteriores, HTML5 incluye etiquetas adicionales que permiten crear sitios web más modernos y compatibles con las tecnologías digitales actuales. [34]
- **CSS:** CSS es el acrónimo de “Cascading Style Sheets”, que significa “Hojas de estilo en cascada”. CSS es un lenguaje declarativo que se utiliza para determinar el formato y diseño visual de las páginas web. CSS permite separar el contenido estructural de una página web (definido en HTML) de su diseño y apariencia visual, permitiendo a los desarrolladores controlar la presentación y el estilo de los elementos de un documento HTML. [35]
- **Javascript:** JavaScript es un lenguaje de programación de alto nivel, interpretado y multiplataforma. Es comúnmente utilizado en el desarrollo web para dar interactividad y dinamismo a las páginas web. Originalmente, JavaScript fue creado para ser ejecutado en el navegador web del cliente, pero actualmente también puede ejecutarse en servidores. Este lenguaje, en el lado del cliente o frontend, permite a los desarrolladores web manipular el contenido HTML y CSS de una página web, responder a eventos del usuario, realizar operaciones matemáticas, interactuar con servicios web, entre otras muchas más opciones. [36]

3.5.- TECNOLOGÍAS BACKEND

Las tecnologías de backend forman la capa de la aplicación encargada de las operaciones del lado del servidor, incluyendo la lógica, el procesamiento de datos y la gestión de la base de datos. En este proyecto se han utilizado las tecnologías backend que se describen a continuación.

3.5.1.- Java

Java es un lenguaje de programación utilizado para crear software compatible con una gran diversidad de sistemas operativos, orientado a objetos y diseñado para tener la menor cantidad posible de dependencias de implementación. Fue creado en 1991 y lanzado en 1995 por James Gosling de Sun Microsystems con la filosofía de "*escribir una vez, ejecutar en cualquier lugar*" (WORA, por sus siglas en inglés). Este lenguaje tiene la particularidad de ser compilado e interpretado al mismo tiempo; esto significa que es un lenguaje simplificado que convierte automáticamente el código en instrucciones de máquina. Actualmente es propiedad de Oracle Corporation [37]. Algunas de sus características más importantes son:

- Orientado a objetos: Java utiliza el paradigma de la programación orientada a objetos, lo que permite modularidad y una significativa reutilización del código.
- Plataforma independiente: Gracias a la Máquina Virtual de Java (JVM), los programas Java pueden ejecutarse en cualquier sistema operativo sin necesidad de modificar el código fuente.
- Sintaxis similar a C++: La sintaxis de Java es similar a C++, facilitando su aprendizaje para programadores familiarizados con C o C++.
- Recolección de basura: Java gestiona automáticamente la memoria mediante la recolección de basura, eliminando objetos innecesarios y liberando memoria.
- Seguridad: La JVM proporciona un entorno controlado con varias características de seguridad, como la verificación de bytecode y la gestión de permisos.
- Multihilo (Multithreading): Java permite el desarrollo de aplicaciones multihilo, posibilitando la ejecución simultánea de varias tareas dentro de un mismo programa.

- Bibliotecas amplias: Java cuenta con una extensa colección de bibliotecas estándar que ofrecen funcionalidades reutilizables para diversas tareas comunes.
- Desarrollo de aplicaciones distribuidas: Java es ampliamente utilizado en el desarrollo de aplicaciones distribuidas y es el lenguaje preferido para muchas aplicaciones empresariales utilizando J2EE (Java 2 Platform, Enterprise Edition).



Figura 3.14: Imagen java.

El ecosistema de Java incluye varias ediciones y herramientas [38]:

- Java SE (Standard Edition): Proporciona el núcleo del lenguaje y las bibliotecas estándar.
- Java EE (Enterprise Edition): Incluye herramientas y bibliotecas para el desarrollo de aplicaciones empresariales robustas y escalables.
- Java ME (Micro Edition): Destinada al desarrollo de aplicaciones para dispositivos móviles y embebidos.

Java ha evolucionado constantemente desde su lanzamiento y sigue siendo uno de los lenguajes de programación más populares y utilizados en la industria, especialmente en el desarrollo de aplicaciones empresariales, aplicaciones web, y sistemas embebidos.

3.5.2.- JSP (JavaServer Pages)

JSP (JavaServer Pages) es una tecnología desarrollada por Sun Microsystems que permite a los desarrolladores crear páginas web dinámicas y basadas en el lenguaje de programación Java, facilitando la generación de contenido HTML de manera dinámica mediante la inserción de código Java directamente en las páginas HTML. Esta tecnología es particularmente útil en el contexto de aplicaciones web empresariales, ya que permite una integración estrecha con bases de datos, servicios web y otros componentes del servidor, aprovechando toda la potencia y robustez de la plataforma Java.

A diferencia de los servlets, que son programas Java que generan páginas web completas, las JSPs permiten a los desarrolladores mezclar directamente el contenido estático, como el HTML, con el contenido dinámico, generado mediante fragmentos de código Java, en una misma página. Esto se logra a través de una serie de etiquetas y expresiones que son interpretadas por el servidor de aplicaciones para producir el contenido dinámico requerido por la aplicación web.

Una de las características más destacadas de JSP es su capacidad para separar claramente el diseño y la lógica de la aplicación; esto se consigue mediante el uso de etiquetas de acción y bibliotecas de etiquetas personalizadas (tag libraries), que encapsulan la lógica de la aplicación en componentes reutilizables y fáciles de mantener. Además, JSP se integra de manera nativa con otras tecnologías Java, como JavaBeans y Enterprise JavaBeans (EJB), lo que permite una arquitectura modular y escalable.[39]

Cuando un navegador solicita una página JSP, el servidor web compila la página en un servlet; este servlet es luego ejecutado para generar el contenido HTML dinámico, que es enviado de vuelta al navegador. Esta compilación solo ocurre la primera vez que se solicita la página, o cuando el archivo JSP ha sido modificado, lo que optimiza el rendimiento en subsecuentes solicitudes al reducir el tiempo de procesamiento necesario para generar la página.

En términos de flujo de trabajo, se puede escribir una página JSP utilizando elementos como directivas, scriptlets y expresiones; las directivas configuran aspectos generales de la página, como la importación de bibliotecas de clases; los scriptlets permiten la inclusión de bloques de código Java, mientras que las expresiones insertan valores directamente en el contenido HTML. Este enfoque flexible y poderoso permite a los desarrolladores crear aplicaciones web dinámicas de manera eficiente, aprovechando toda la gama de funcionalidades que ofrece la plataforma Java.

Las directivas disponibles son [40]:

- include: Incluye el contenido de un fichero en la página mediante el atributo file.
`<%@ include file="cabecera.html" %>`
- taglib: Importa bibliotecas de etiquetas (Tag Libraries)
`<%@ taglib uri="/tags/struts-html" prefix="html" %>`
- page: Especifica atributos relacionados con la página a procesar. Dichos atributos se muestran en la tabla 3.1.

Tabla 3.1: Directivas de JSP.

Atributo	Sintaxis	Utilización
import	<%@ page import="class; class" %>	Importa clases y paquetes Java para ser utilizados dentro del fichero JSP.
session	<%@ page session="false" %>	Especifica si utiliza los datos contenidos en sesión; por defecto "true".
contentType	<%@ page contentType="class; class" %>	Especifica el tipo MIME del objeto "response"; por defecto "text/html;charset=ISO-8859-1"
buffer	<%@ page buffer="12KB" %>	Buffer utilizado por el objeto writer "out"; puede tomar el valor de "none"; por defecto "8KB".
errorPage	<%@ page errorPage="/path_to_error_page" %>	Especifica la ruta de la página de error que será invocada en caso de producirse una excepción durante la ejecución de este fichero JSP.
isErrorPage	<%@ page isErrorPage="true" %>	Determina si este fichero JSP es una página que maneja excepciones. Únicamente a este tipo de páginas pueden acceder a la variable implícita "exception", que contiene la excepción que provocó la llamada a la página de error.

Por último, cabe destacar que JSP se complementa con otras tecnologías y frameworks, como Servlets, JDBC (Java Database Connectivity) para el acceso a bases de datos, y frameworks MVC (Modelo-Vista-Controlador) como Struts y Spring MVC, los cuales facilitan aún más el desarrollo de aplicaciones web robustas, mantenibles y escalables.

3.5.3.- SQL (Structured Query Language)

El lenguaje de consulta estructurada (SQL) es un lenguaje de programación utilizado para gestionar y manipular información en bases de datos relacionales. Estas bases de datos organizan la información en tablas, con filas y columnas que representan distintos atributos y las relaciones entre los datos. Con SQL, puedes realizar operaciones como almacenar, actualizar, eliminar, buscar y recuperar datos en la base de datos. También se puede usar SQL para mantener y optimizar el rendimiento de la base de datos [41].

SQL es un lenguaje que aprovecha la flexibilidad de los sistemas relacionales para realizar una amplia gama de operaciones. Es declarativo y de alto nivel, lo cual, gracias a su base teórica sólida y enfoque en conjuntos de registros, facilita una alta productividad en la codificación y la orientación a objetos. Esto significa que una sola sentencia en SQL puede ser equivalente a varios programas en lenguajes de bajo nivel que trabajan con registros individuales. Además de lo anterior SQL también tiene las siguientes características [42]:

- **Lenguaje de definición de datos (LDD):** SQL proporciona comandos para crear, eliminar y modificar la estructura de las bases de datos, incluyendo tablas, claves, índices y relaciones.
- **Lenguaje de manipulación de datos (LMD):** Utiliza álgebra relacional y cálculo relacional de tuplas para realizar consultas y extraer datos. Incluye comandos para insertar, actualizar y borrar datos.
- **Autorización:** SQL incluye comandos para gestionar los permisos y los derechos de acceso a los datos, asegurando que solo usuarios autorizados puedan realizar ciertas operaciones.
- **Integridad:** SQL permite definir reglas de integridad para asegurarse de que los datos en la base de datos sean correctos y coherentes.
- **Definición de vistas:** Permite crear vistas, que son formas personalizadas de ver y manipular datos sin afectar las tablas subyacentes.
- **Control de transacciones:** Proporciona comandos para especificar el comienzo y el final de una transacción.
- **SQL incorporado y dinámico:** Se pueden incorporar instrucciones de SQL en lenguajes de programación como C++, Java, PHP, y más, permitiendo la creación de aplicaciones que interactúan con bases de datos.

Estas características hacen de SQL una herramienta esencial para trabajar con bases de datos relacionales, ofreciendo flexibilidad, eficiencia y seguridad en el manejo de datos.

Capítulo 4

Metodología y resultados

4.1.- PLANIFICACIÓN DEL PROYECTO

El ciclo de vida del desarrollo del software contempla las fases necesarias para validar el desarrollo del software y así garantizar que este cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo, asegurándose de que los métodos usados son apropiados. Es el proceso que se sigue para construir, entregar y hacer evolucionar el software, desde que se tiene una idea, hasta la entrega del sistema funcional al cliente final; dicho proceso incluye también el mantenimiento y evolución del sistema posterior, e incluso su retirada cuando haya quedado obsoleto [43].

En este proyecto usaremos el ciclo de vida en cascada, en este modelo cada una de estas fases se ejecuta tan solo una vez (en principio, salvo retroalimentación). Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente. De este modo, sólo cuando una fase termine se podrá continuar con la siguiente, y así progresivamente.

A continuación, se detallan brevemente cada una de las fases del ciclo de vida en cascada[44]:

- **Planificación:** Las tareas iniciales que se realizarán en esta fase inicial del proyecto incluyen actividades tales como la determinar del ámbito del proyecto, la realización de un estudio de viabilidad, el análisis de los riesgos asociados al proyecto, una estimación del coste del proyecto, su planificación temporal y la asignación de recursos a las distintas etapas del proyecto.
- **Análisis:** Lo primero que debemos hacer para construir un sistema de información es averiguar qué es exactamente lo que tiene que hacer el sistema. La etapa de análisis en el ciclo de vida del software corresponde al proceso mediante el cual se intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer).

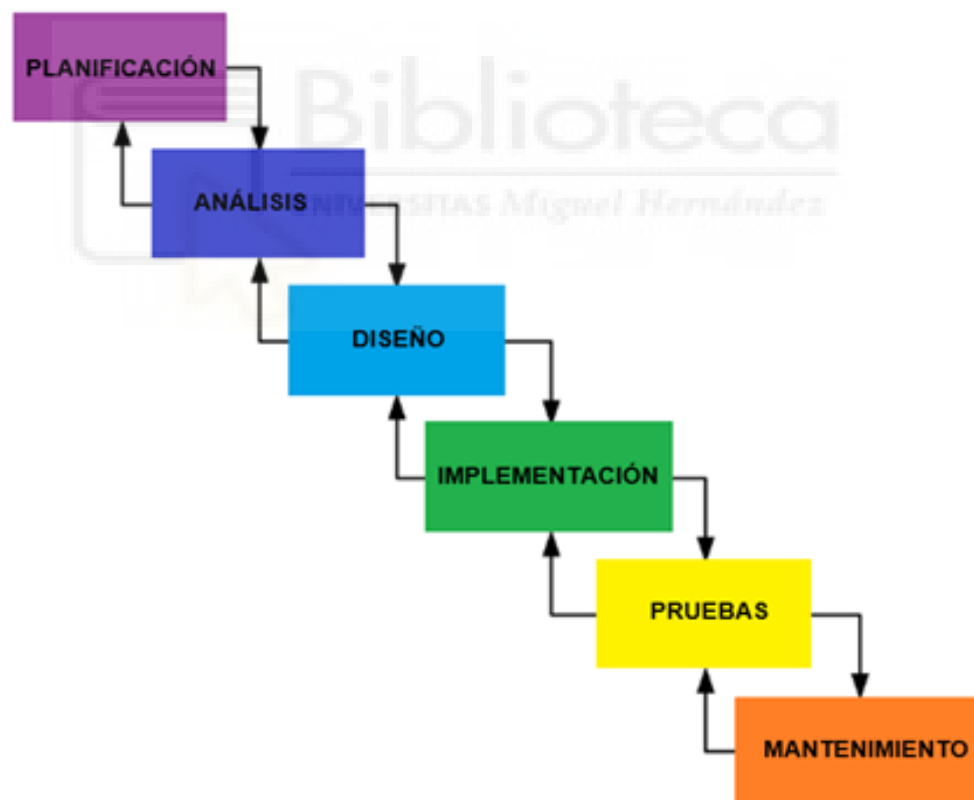


Figura 4.1: Ciclo de vida en cascada

- **Diseño:** Mientras que los modelos utilizados en la etapa de análisis representan los requisitos del usuario desde distintos puntos de vista (el qué), los modelos que se utilizan en la fase de diseño representan las características del sistema que nos permitirán implementarlo de forma efectiva (el cómo).

- Implementación: Una vez que sabemos qué funciones debe desempeñar nuestro sistema de información (análisis) y hemos decidido cómo vamos a organizar sus distintos componentes (diseño), es el momento de pasar a la etapa de implementación, pero nunca antes de escribir una sola línea de código (o de crear una tabla en nuestra base de datos) es fundamental haber comprendido bien el problema que se pretende resolver y haber aplicado principios básicos de diseño que nos permitan construir un sistema de información de calidad. Para la fase de implementación hemos de seleccionar las herramientas adecuadas, un entorno de desarrollo que facilite nuestro trabajo y un lenguaje de programación apropiado para el tipo de sistema que vayamos a construir. La elección de estas herramientas dependerá en gran parte de las decisiones de diseño que hayamos tomado hasta el momento y del entorno en el que nuestro sistema deberá funcionar.
- Pruebas: Errar es humano y la etapa de pruebas tiene como objetivo detectar los errores que se hayan podido cometer en las etapas anteriores del proyecto (y, eventualmente, corregirlos). Lo ideal, además, es hacerlo antes de que el usuario final del sistema los tenga que sufrir. De hecho, una prueba es un éxito cuando se detecta un error (y no al revés, como intuitivamente se podría pensar).
- Mantenimiento: Una vez concluidas las etapas de desarrollo de un sistema de información (análisis, diseño, implementación y pruebas), llega el instante de poner el sistema en funcionamiento y con él su mantenimiento. En esta fase además de la eliminación de defectos, se incluye principalmente cómo añadir una nueva funcionalidad a un sistema ya existente, el mantenimiento repite "en miniatura" el ciclo de vida completo de un sistema de información. Es más, a las tareas normales de desarrollo hemos de añadirle una nueva, comprender el sistema que ya existe, por lo que se podría decir que el mantenimiento de un sistema es más difícil que su desarrollo.[45]

Una forma de llevar la planificación de un proyecto es mediante un diagrama de Gantt, que proporciona una vista general de las tareas programadas, todas las partes implicadas sabrán qué tareas tienen que completarse y en qué fecha. Un diagrama de Gantt muestra:

- La fecha de inicio y finalización del proyecto
- Qué tareas (y subtareas) forman el proyecto y sus fechas estimadas de inicio y fin
- Cómo se superponen las tareas y/o si hay alguna relación entre ellas

EDT	Nombre de tarea / Título	Fecha inicio	Fecha fin	Duración (días)	Predecesor
1	CATERING CHAPATA	28/02/2024	18/06/2024	80	
1.1	DOCUMENTACIÓN	28/02/2024	19/03/2024	15	
1.1.1	Planificación de proyecto	28/02/2024	28/02/2024	1	
1.1.2	Captura de requisitos	29/02/2024	18/03/2024	13	
1.1.2.1	Requisitos funcionales	29/02/2024	06/03/2024	5	
1.1.2.2	Roles de usuario	07/03/2024	08/03/2024	2	
1.1.2.3	Casos de uso	11/03/2024	18/03/2024	6	1.1.2.2
1.1.3	Diseño	07/03/2024	19/03/2024	9	
1.1.3.1	Base de datos	07/03/2024	12/03/2024	4	1.1.2.1
1.1.3.2	Interfaz grafica	13/03/2024	19/03/2024	5	
1.2	DESARROLLO APP	20/03/2024	18/06/2024	65	1.1
1.2.1	Usuario no registrado	20/03/2024	15/04/2024	19	
1.2.1.1	Modulo de inicio	20/03/2024	22/03/2024	3	
1.2.1.2	Modulo de productos	25/03/2024	28/03/2024	4	
1.2.1.3	Modulo de personal	29/03/2024	03/04/2024	4	
1.2.1.4	Modulo de finca	04/04/2024	09/04/2024	4	
1.2.1.5	Modulo de contacto	10/04/2024	10/04/2024	1	
1.2.1.6	Modulo login	11/04/2024	15/04/2024	3	
1.2.2	Usuario registrado	16/04/2024	06/05/2024	15	1.2.1
1.2.2.1	Modulo de carrito	16/04/2024	24/04/2024	7	
1.2.2.2	Modulo de reservas	25/04/2024	01/05/2024	5	
1.2.2.3	Modulo de perfil	02/05/2024	06/05/2024	3	
1.2.3	Administrador	07/05/2024	11/06/2024	26	1.2.2
1.2.3.1	Modulo de configuración	07/05/2024	27/05/2024	15	
1.2.3.1.1	Configuración de carousel	07/05/2024	08/05/2024	2	
1.2.3.1.2	Configuración de reseñas	09/05/2024	10/05/2024	2	
1.2.3.1.3	Configuración de productos	13/05/2024	16/05/2024	4	
1.2.3.1.4	Configuración de personal	17/05/2024	22/05/2024	4	
1.2.3.1.5	Configuración de fincas	23/05/2024	27/05/2024	3	
1.2.3.2	Modulo de gestión de reservas	28/05/2024	04/06/2024	6	
1.2.3.3	Modulo de gestión de usuarios	05/06/2024	11/06/2024	5	
1.2.4	Pruebas/implementación	12/06/2024	18/06/2024	5	1.2
1.2.4.1	Pruebas de componentes	12/06/2024	13/06/2024	2	
1.2.4.2	Pruebas de integración	14/06/2024	18/06/2024	3	

Figura 4.2: Planificación de tareas y fechas

La figura 4.2 muestra las diferentes tareas realizadas en este proyecto así como la dirección de las mismas (con sus fechas de inicio y fin). En la figura 4.3 se vuelven a mostrar las tareas del proyecto y la línea temporal de cada una de ellas, representando el comienzo y fin de cada una de ellas y sus dependencias.

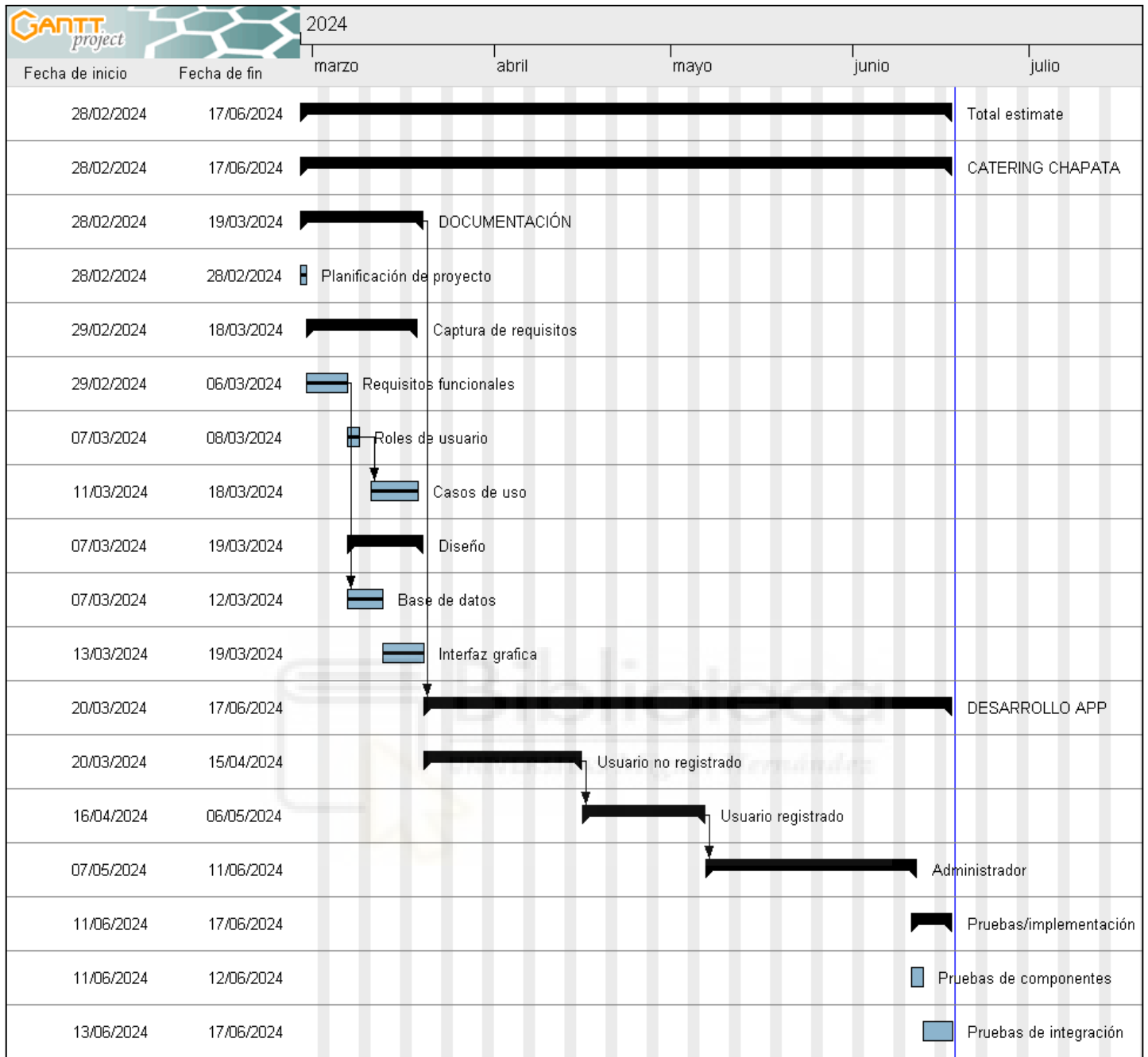


Figura 4.3: Planificación de tareas línea temporal

4.2.- CAPTURA DE REQUISITOS

4.2.1.- Requisitos funcionales

La aplicación a desarrollar debe ser capaz de realizar las siguientes tareas o funciones, que se corresponden con los objetivos del proyecto y, en su momento constituirán cada una de las opciones de los menús de la aplicación:

Inicio: Esta es la página principal de la webapp, sirve como el punto de entrada principal para los usuarios, desde aquí podremos ver un carrusel de imágenes y una sección de reseñas de usuarios de los productos o servicios ofrecidos. También veremos el menú superior desde el que se tiene acceso a las áreas "Productos", "Personal", "Finca", "Contacto", "Carrito", "Usuario" y, si se está registrado como administrador, también estarán las opciones de "Administrador" "Reservas" y "Usuarios".

Producto: Desde aquí se pueden ver y explorar diferentes categorías de productos, como "Principal", "Entrante", "Postre" y "Bebidas". Cada categoría se presenta de forma ordenada en la página. Se pueden ver detalles de los productos, como nombre, descripción, precio y la opción añadir al carrito. Esta última funcionalidad sería solo para los usuarios registrados.

Personal: Proporcionar información sobre el personal disponible, incluyendo detalles como profesión (especialidad), disponibilidad y tarifas.

Finca: Mostrar a los usuarios explorar las diferentes fincas disponibles, que pueden ser utilizadas para la organización de eventos, dando de cada una de ellas información detallada incluyendo ubicación, capacidad y servicios ofrecidos.

Contacto : Mostrar a los usuarios la ubicación de la empresa en un mapa y proporcionar un formulario de contacto para enviar mensajes a la empresa.

Carrito: Mostrar los productos, el personal y las fincas seleccionados por el usuario para su compra, además de permitir al usuario gestionar las unidades de los productos en el carrito.

Usuario: Ver, editar y actualizar los distintos datos del usuario para que queden registrados en la base de datos. También puede ver su historial de reservas pudiendo editar y actualizar cualquier evento que no esté realizado o confirmado con las condiciones actuales.

Administrador: Permitir al administrador gestionar la configuración de la web:

- Gestión del carrusel de imágenes: Permite al administrador agregar, editar y eliminar imágenes para el carrusel principal de la aplicación.
- Gestión de reseñas: Permite al administrador añadir, editar y eliminar reseñas de usuarios sobre productos, servicios o experiencias proporcionadas.
- Gestión de productos: Permite al administrador agregar, editar y eliminar productos disponibles en la plataforma, incluyendo información detallada como descripción, precio y disponibilidad.

- **Gestión del personal:** Permite al administrador gestionar el personal de la aplicación, pudiendo añadir, editar y eliminar información de las personas así como sus profesiones, experiencia y tarifas
- **Gestión de fincas:** Permite al administrador agregar, editar y eliminar información relacionada con las fincas asociadas a la aplicación, incluyendo detalles de ubicación, servicios ofrecidos y fechas de disponibilidad.

Reservas: Permite al administrador revisar, confirmar o rechazar las reservas realizadas por los usuarios, manteniendo un registro actualizado del estado de cada reserva.

Usuarios: Permite al administrador gestionar las cuentas de usuario registradas en la aplicación, incluyendo la capacidad de crear, editar y eliminar cuentas según sea necesario.

4.2.2.- Roles de usuario

En el punto anterior se han descrito los requisitos funcionales, estos requisitos serán opciones del menú de la aplicación web. Habrá diferentes roles de usuarios, cada rol tendrá unas atribuciones, por ello al identificarse dependiendo del su rol tendrá unas opciones del menú u otras. A continuación, se describen los roles de usuario de la aplicación:

Tabla 4.1: Rol usuario no registrado.

Usuario	<i>Usuario no registrado</i>
Descripción	El usuario no registrado podrá registrarse o bien navegar por la aplicación viendo los productos disponibles, el personal, la finca, contactar con la empresa y ver su ubicación, pero sin poder reservar ninguno de los servicios que aparecen en ella.
Casos de uso	<i>CU-1, CU-2, CU-3, CU-4, CU-5, CU-6, CU-7, CU-8, CU-9, CU-10, CU-11</i>

Tabla 4.2: Rol usuario registrado.

Usuario	<i>Usuario registrado</i>
Descripción	El usuario registrado (he identificado) es aquel que dispone de privilegios de reserva, con lo cual iniciando sesión podrá reservar cualquiera de los servicios ofrecidos.
Casos de uso	<i>CU-11, CU-12, CU-13, CU-14, CU-15, CU-16, CU-17, CU-18, CU-19, CU-20, CU-21, CU-22, CU-23, CU-24, CU-25, CU-26</i>

Tabla 4.3: Rol administrador.

Usuario	<i>Administrador</i>
Descripción	El usuario con rol de administrador será el que gestione toda la aplicación. Podrá añadir, modificar o eliminar, imágenes del carrusel, reseñas de clientes, modificar cualquier producto, personal o finca. En cuanto a la gestión de usuario el administrador podrá dar de alta, modificar o eliminar usuarios de la aplicación.
Casos de uso	<i>Todos desde el CU-1 hasta el CU-52</i>

Al tener varios tipos de usuarios (roles) en una aplicación, suelen darse relaciones de herencia. Esto quiere decir que entre un rol y otro se comparten algunas funcionalidades, la siguiente figura muestra la relación de herencia entre nuestros usuarios.

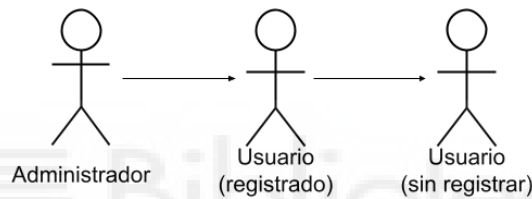


Figura 4.4: Diagrama jerárquico de los usuarios

4.2.3.- Casos de uso

El primer diagrama de casos de uso es el del usuario no registrado, es el más sencillo ya que solo puede visualizar contenido de la aplicación, además de registrarse e identificarse.

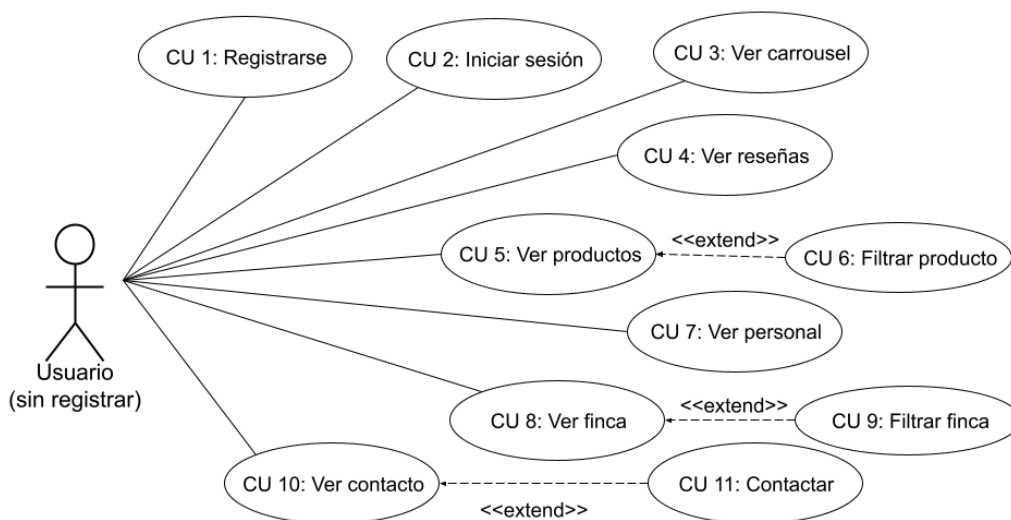


Figura 4.5: Diagrama casos de uso del usuario no registrado

El usuario registrado hereda la funcionalidad del usuario no registrado, es decir, además de su propios casos de uso, también puede realizar los del usuario no registrado (figura 4.5).

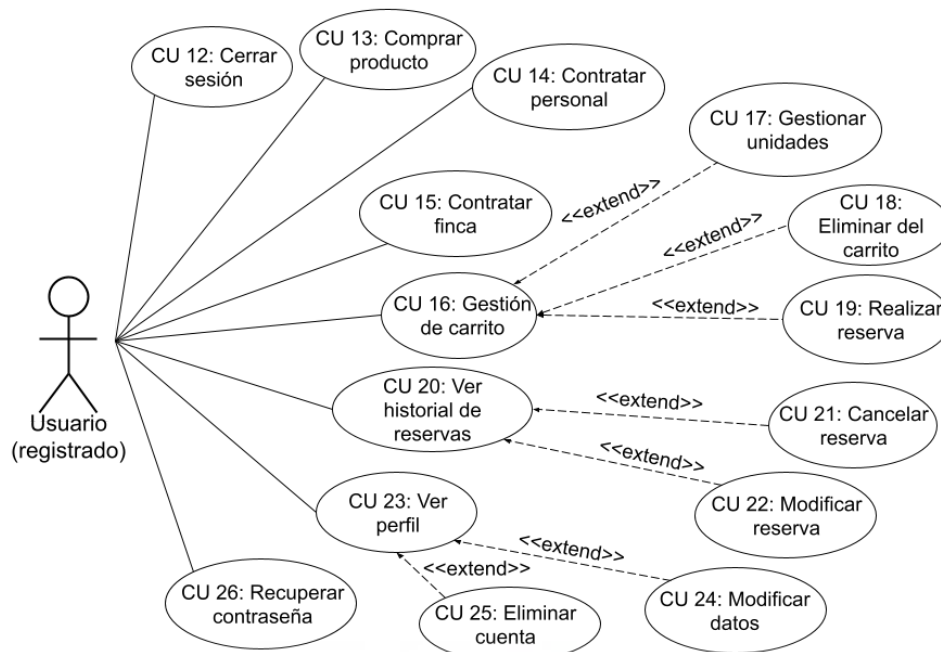


Figura 4.6: Diagrama casos de uso del usuario registrado

Por último está el usuario con rol de administrador, el usuario con más permisos dentro de la aplicación, que hereda funcionalidades del usuario registrado y del no registrado.

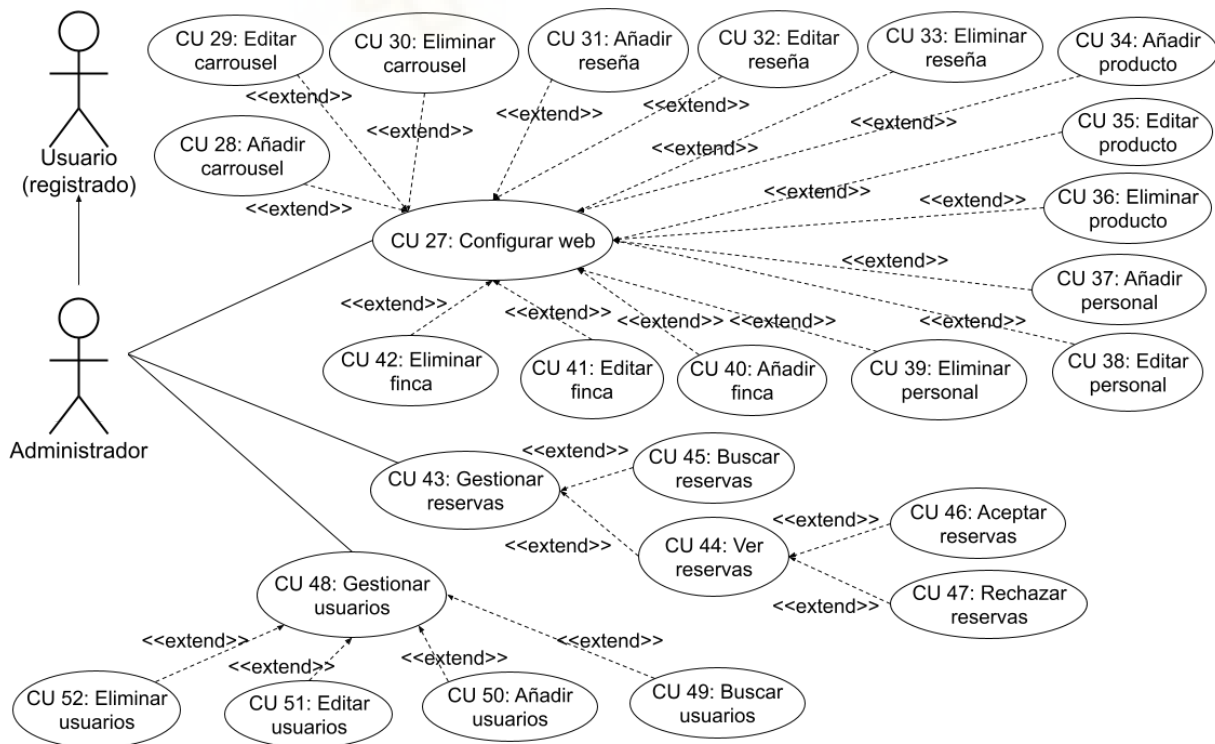


Figura 4.7: Diagrama casos de uso del administrador

En las tablas incluidas en el anexo I de este proyecto se proporcionará el detalle de cómo debe ser el funcionamiento de los diferentes casos de usos cuando los usuario interactúen con la aplicación.

4.3.- DISEÑO

En este apartado se va a describir cómo tiene que ser algunos de los elementos de la aplicación como la base de datos, las clases que habrá que implementar, o el aspecto que se espera que tenga la interfaz de usuario.

4.3.1.- Base de datos

Para almacenar los datos que requiere la aplicación se ha diseñado una base de datos relacional con las siguientes tablas:

- Evento: Registra los eventos que serían las celebraciones organizadas por cada cliente. Estos eventos están compuestos por las contrataciones, menús y fincas.
- Contratación: Registra las contrataciones de personal para los eventos.
- Menú: Almacena los menús utilizados en los eventos.
- Fincas: Almacena información sobre las fincas disponibles para los eventos.
- Personal: Almacena información sobre el personal disponible para la contratación en eventos.
- Producto: Almacena información sobre los productos disponibles para la elaboración de los menús.
- Usuarios: Almacena información sobre los usuarios registrados en la aplicación.
- Perfiles: Almacena información sobre los perfiles de los usuarios registrados.
- Carrousel: Almacena información sobre las imágenes utilizadas en el carrusel del slider de la página principal.
- ClienteReviews: Almacena reseñas y opiniones de los clientes.

En la figura 4.8 se muestra el modelo Entidad/Relación de la base de datos que finalmente se ha diseñado para la aplicación.

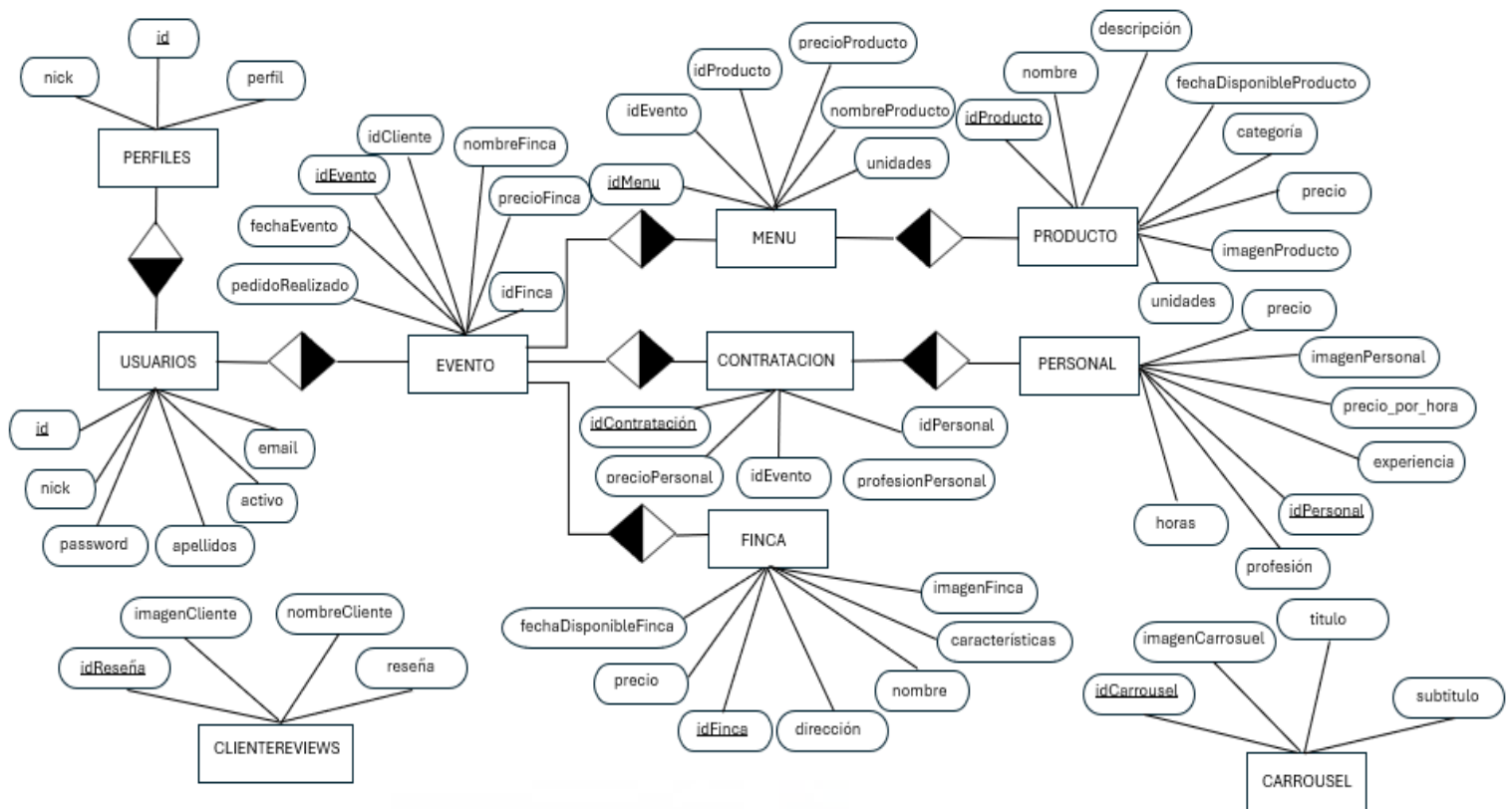


Figura 4.8: Diseño base de datos

4.3.2.- Diagrama de clase

Se ha decidido incluir el diagrama de clases en esta documentación debido a que la aplicación está desarrollada con Java, que es un lenguaje de programación orientado a objetos. En la figura 4.9 se muestran las clases que se han implementado de forma esquemática (por razón de espacio). En el anexo II se puede consultar una descripción más detallada que incluye las propiedades y métodos de cada clase.

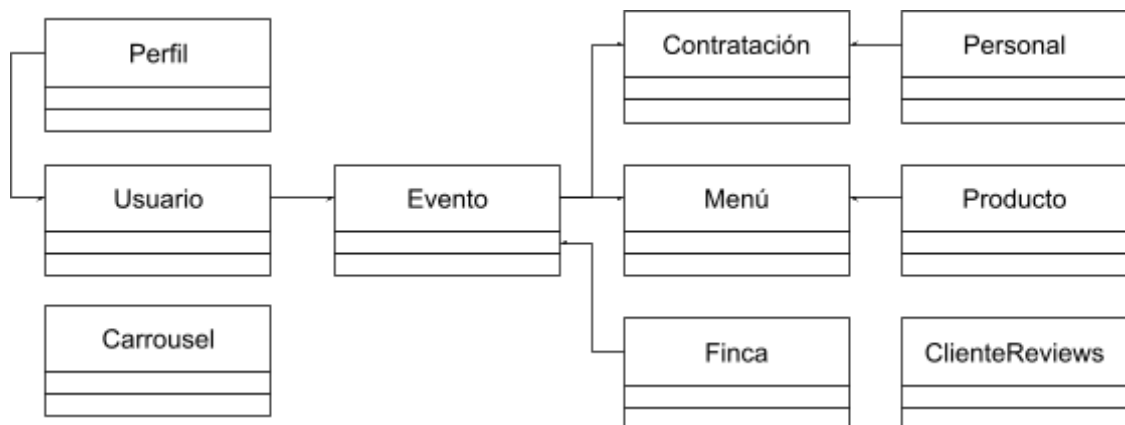


Figura 4.9: Diagrama de clases

4.3.3.- Esbozo

A continuación se muestra lo que se pensó y se planteó como primer diseño de lo que iba a ser la interfaz gráfica de la aplicación. Aunque han cambiado algunos detalles, a grandes rasgos se ha respetado bastante.

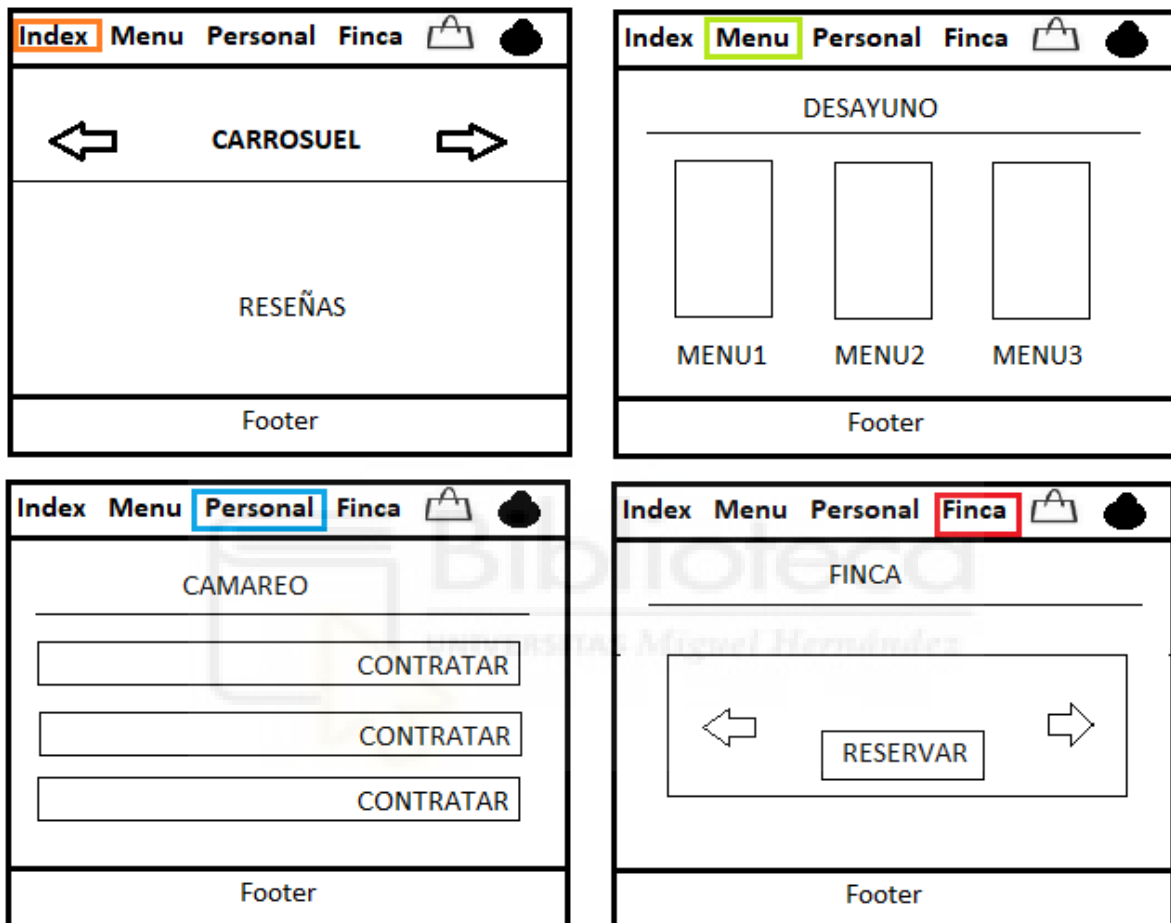


Figura 4.10: Primer diseño de interfaz de la aplicación.

La figura 4.10 muestra el diseño de las opciones “Index” (home), “Menú”, “Personal” y “Finca”.

4.4.- Implementación

En este apartado se va a mostrar el resultado final de la aplicación desarrollada. Destacar que el presente trabajo es un proyecto meramente académico, su finalidad es únicamente para el desarrollo de este trabajo de fin de grado, en principio no está previsto su puesta en marcha en un entorno de producción.

En la vista de la figura 4.11 tenemos el home (o index) de la aplicación. Esta página se muestra igual para todos los usuarios, donde se puede ver el carrusel de imágenes y el slider de reseñas de clientes.

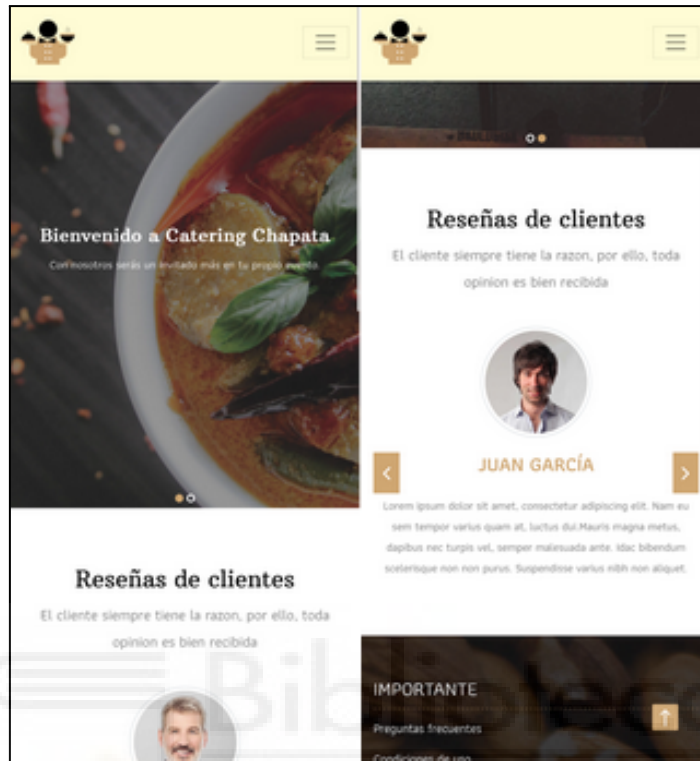


Figura 4.11: Página principal.

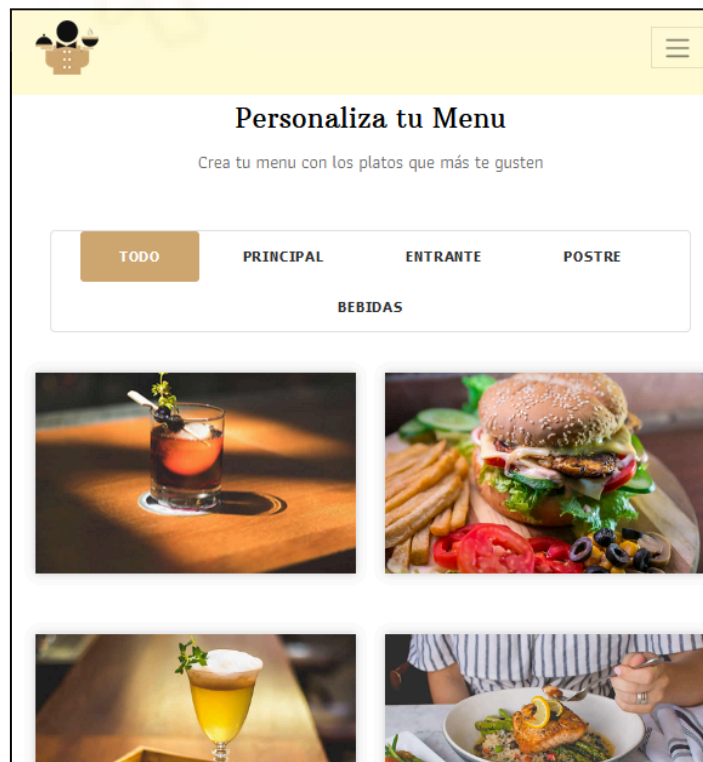


Figura 4.12: Página de productos

En la figura 4.12 se muestran los productos disponibles en la aplicación. Esta página se muestra igual para todos los usuarios, se pueden ver los productos con su nombre, una imagen, su descripción y el precio, pero únicamente podrá añadirlos al carrito el usuario que previamente se haya identificado.

En la figura 4.13 tenemos las distintas profesiones (especialidades) disponibles para contratar en la aplicación. Esta página se muestra igual para todos los usuarios, se muestra el personal con su precio por hora y las horas disponibles, pero únicamente podrá contratar el servicio si el usuario está logueado en la aplicación.

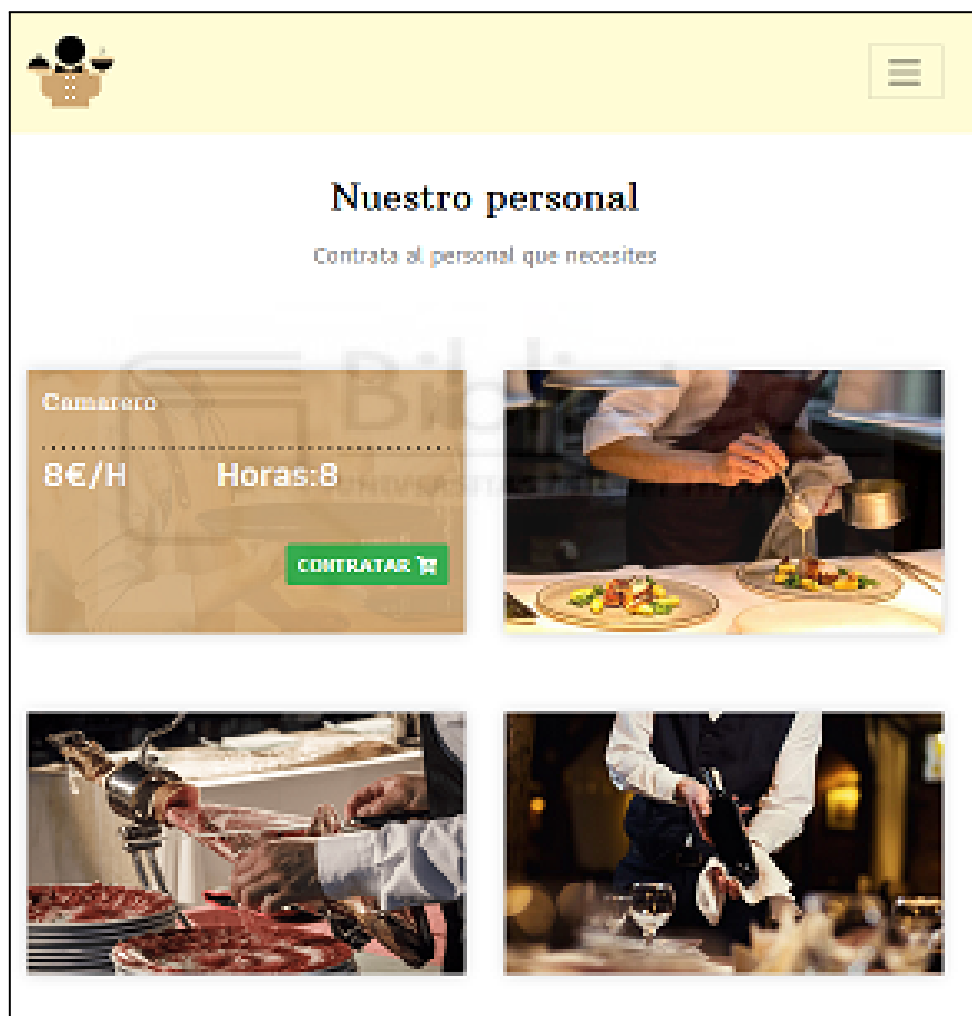


Figura 4.13: Página de personal.

En la figura 4.14 se muestra la página de fincas a la izquierda, donde si filtramos por fecha obtendremos las distintas fincas disponibles para reservar en la aplicación para la fecha seleccionada. Esta página se muestra igual para todos los usuarios, se muestra una imagen de la finca, una descripción de ella y su precio, pero únicamente podrá reservar si el usuario está logueado en la aplicación.

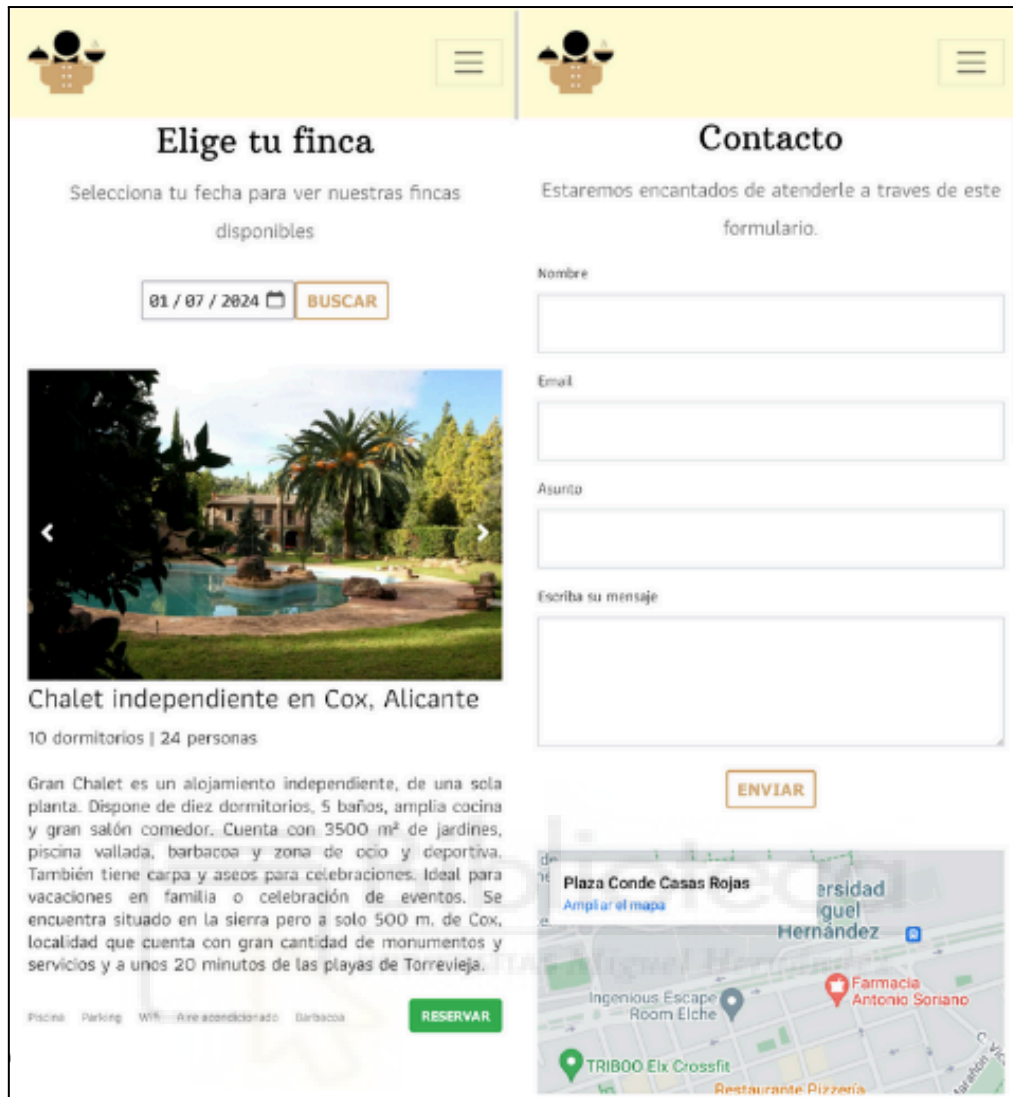


Figura 4.14: Página de finca y página de contacto.

En la parte derecha de la figura 4.14 está la página de contacto, donde hay un formulario de contacto y un mapa de ubicación de la aplicación.

Para tener acceso a las operaciones que se van a describir a continuación es necesario que el usuario esté identificado con el rol de cliente para poder realizar reservas; para la parte de gestión de la aplicación, el usuario tendrá que estar identificado con el rol de administrador.

En la figura 4.15 se muestran los tres diferentes menús superiores de los que dispone la aplicación en función del rol del usuario, por orden, la parte izquierda es lo que verá el usuario no registrado, la parte central es lo que verá el usuario registrado como cliente, donde, a diferencia del usuario no registrado, vemos que dispone de un submenú donde tiene toda la información sobre su perfil y sus reservas realizadas. Por último, en la parte derecha se encuentra el menú disponible para el administrador, donde a diferencia del usuario registrado como cliente, vemos la pestaña de administrador para la configuración

de la aplicación, la pestaña de reservas para la gestión de las reservas y por ultimo usuarios, para todo el control de los usuario de la aplicación.

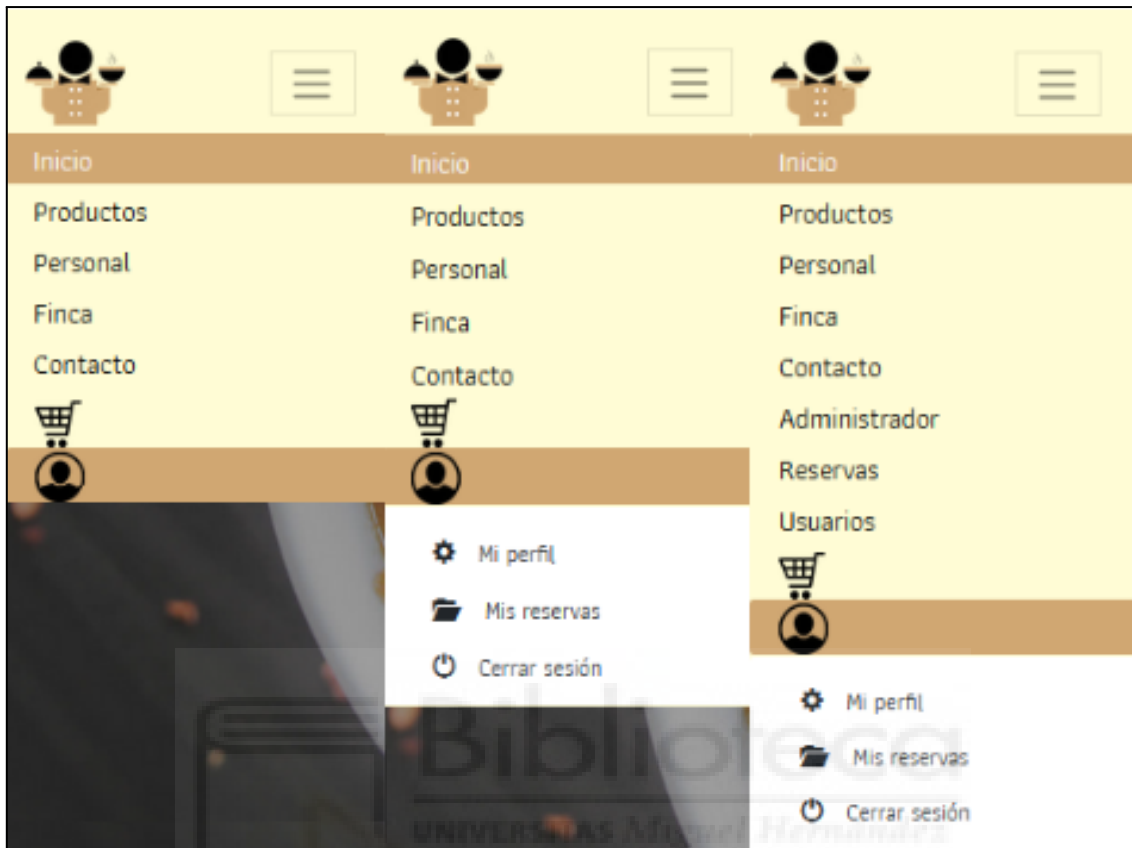


Figura 4.15: Distintos header.

En la siguiente figura 4.16, se muestra como se vería el carrito con los distintos elementos a reservar de los que dispone la aplicación. Todo evento debe contener una fecha antes de realizar la reserva, salvo en caso de seleccionar una finca, en cuyo caso esa será la fecha de la reserva. Como el control de unidades se realiza en el carrito, tanto para productos como para el personal, se dispone de un botón para incrementar y decrementar unidades. También se dispone de un botón de borrado para todos los elementos del carrito, y por último, el botón de confirmar reserva, donde se muestra el precio final de la reserva.

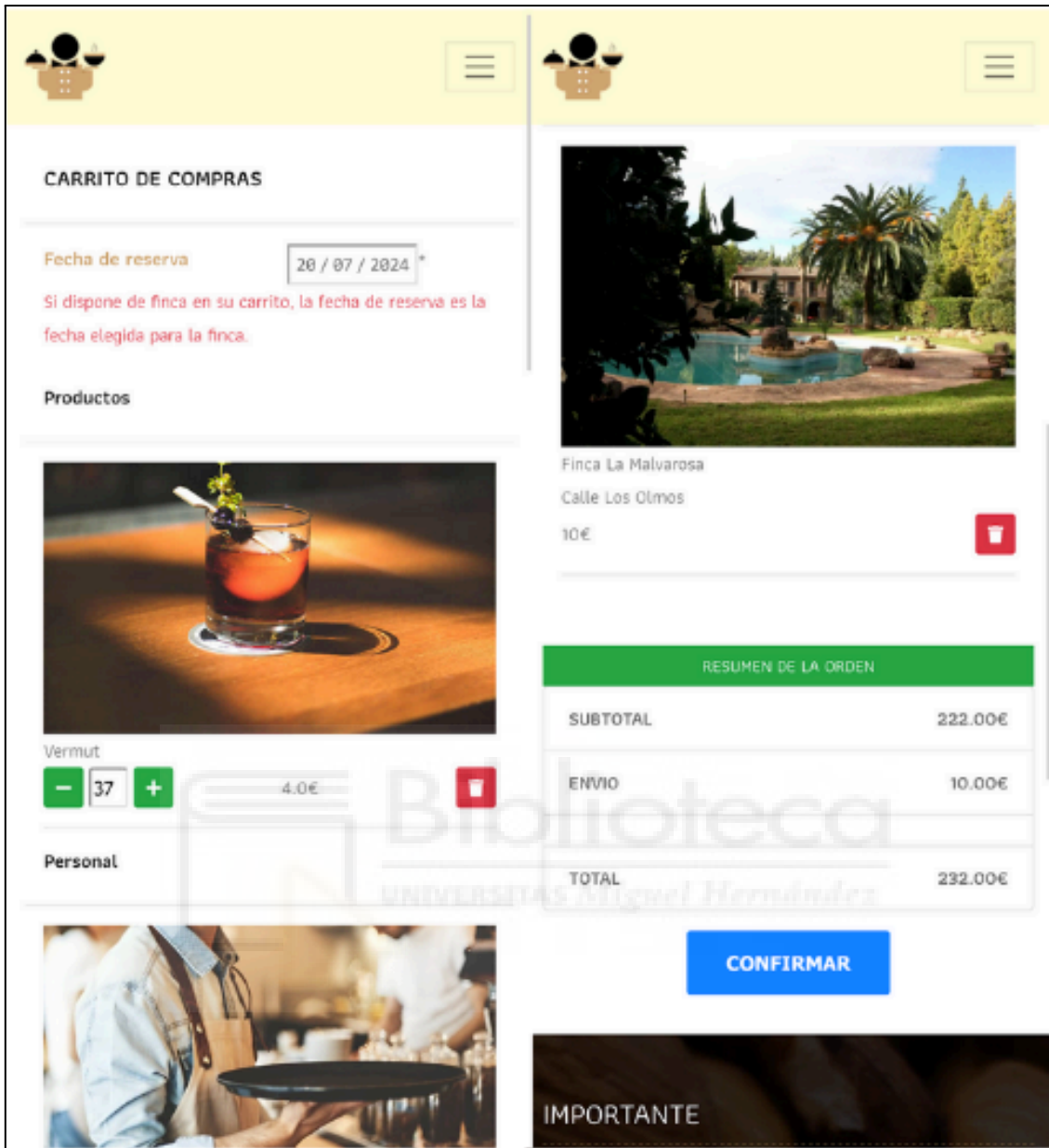


Figura 4.16: Carrito

A continuación se mostrarán las tres opciones en el header que muestra los paneles de “Administrador” (figura 4.17) , “Reservas” (figura 4.18) y “Usuarios” (figura 4.19) de los que dispone el usuario con rol de administrador para gestionar toda la aplicación. Comenzando de izquierda a derecha (figura 4.17), la primera pestaña que nos encontramos es la de administración, desde la cual podremos realizar las acciones de altas, bajas y modificaciones de los elementos que componen la aplicación que son carousel, reviews de clientes, productos, personal y finca.



Figura 4.17: Panel de administrador.

En la siguiente figura 4.18 se muestra la opción de reservas, desde la cual se pueden realizar búsquedas de reservas, por el número de la reserva, por fecha o por el estado de la misma con el objetivo de poder gestionar la reserva deseada. Si la reserva se encuentra en estado “*Pendiente*” es porque el usuario la ha realizado, pero todavía no se le ha confirmado que se le pueda atender. El administrador es el encargado de modificar ese estado, si la reserva va a ser atendida cambiará a estado “*Confirmado*”. Si no es posible atender la reserva el administrador cambiará el estado a “*Cancelado*”.




Gestor de reservas

ID reserva: **FILTRAR** Fecha reserva: **FILTRAR**

Estado: **FILTRAR**

Nick	Nombre	Apellidos	E-mail	Fecha del evento	Nº de reserva	Estado	Ver reserva
vidal	Vidal	Urbaneja	vidal@gmail.com	28/06/2024	227	Pendiente	
vidal	Vidal	Urbaneja	vidal@gmail.com	28/06/2024	228	Confirmado	

Figura 4.18: Panel de reservas.

Para finalizar el apartado sobre implementación, en la figura 4.19 se muestra el panel de usuarios desde el cual el administrador puede realizar búsquedas de usuarios, por nick o por nombre para modificar el usuario deseado o eliminarlo. Además de tener la posibilidad de crear usuarios y de dar y quitar privilegios de administrador.

Gestor de usuarios

Nick: **FILTRAR** Nombre: **FILTRAR**

Nick	Nombre	Apellidos	E-mail	Password	Perfil	Gestion
vidal	Vidal	Urbaneja	vidal@gmail.com	{noop} Vida123*	ADMINISTRADOR	
jon	Juan	Garcia	juan@gmail.com	{noop} Juan123*	CLIENTE	
loreca	Lorena	Castillo Garcia	lorena.castillo03@goumh.umh.es	{noop} Lorena123*	CLIENTE	
martha21	Marta	Molina	marta@gmail.com	{noop} Marta123*	CLIENTE	

NUEVO

Figura 4.19: Panel de usuarios.

4.5.- Pruebas

Se han realizado distintas pruebas en la realización del proyecto con el objetivo de validar la aplicación final:

4.5.1.- Pruebas de interfaz de usuario (UI)

Se han realizado pruebas de interfaz de usuario (UI), también conocidas como pruebas de end-to-end (E2E). Estas pruebas simulan el comportamiento del usuario y verifican que la aplicación responda adecuadamente a las interacciones del usuario, incluyendo la aparición de mensajes emergentes (pop-ups).

Comenzaremos mostrando el popup que muestra la aplicación cuando se intenta añadir un producto o personal que ya está en el carrito, recordemos que esta aplicación realiza la

gestión de unidades en el carrito, por tanto no debe permitir que exista más de un elemento, si se desean más unidades habrá aumentarlas desde el carrito.



Figura 4.20: Pop up control de unidades

También se realizan notificaciones en forma de popup cuando se añade, modifica o elimina un elemento, por ejemplo con un producto:



Figura 4.21: Pop up producto añadido y confirmación de borrado

4.5.2.- Pruebas de validación de formularios.

Se enfocan en verificar la funcionalidad y la retroalimentación de los formularios en la interfaz de usuario asegurando que el formulario maneje adecuadamente las entradas del usuario, mostrando mensajes de error claros y evitando el envío de datos inválidos. Para ello se han realizado las siguientes pruebas:

- Verificar que todos los campos obligatorios sean completados.
- Validar formatos de entrada, como correos electrónicos
- Asegurar que las contraseñas cumplan con los requisitos de seguridad.
- Verificar que se muestran mensajes de error adecuados.

En el formulario de creación de cuenta podemos ver detalladamente cada uno de los puntos anteriores.

Nombre* Apellidos*

Rellene este campo. Email*

El nick ya está en uso El formato del email no es válido




Password*


La contraseña debe contener al menos 8 caracteres, una letra mayúscula, una minúscula, un número y un carácter especial


REGISTRARSE

Figura 4.22: Validación del formulario de registro.

Al igual que el formulario de registro, si hay un intento erróneo de identificación de usuario (id de usuario y/o contraseña incorrectos), esto se indica con un mensaje como el que puede observarse en la figura 4.23.

Sign In   

 Usuario...

 Password...

Recordar

El usuario o la contraseña son incorrectos

LOGIN

¿No tienes cuenta? [Regístrate](#)
¿Olvidaste tu contraseña?

Figura 4.23: Validación del formulario de login.

Para las validaciones de contraseña, email y nick, que son más complejas, se ha creado un paquete dentro de la aplicación con clases dedicadas a crear anotaciones para estas validaciones, un ejemplo de ellas es la que se muestra en la figura 4.24, indica la comprobación para que un email se considere válido, basada en una expresión regular.

```
package lcastillo.app.validator;

import javax.validation.ConstraintValidator;

public class CustomEmailValidator implements ConstraintValidator<CustomEmail, String> {

    @Override
    public void initialize(CustomEmail constraintAnnotation) {
    }

    @Override
    public boolean isValid(String value, ConstraintValidatorContext context) {
        // Logic to validate custom email format
        // Replace with your actual validation logic using regex
        return value.matches("[a-zA-Z0-9._%+-]+@(gmail|hotmail|yahoo|outlook|aol|protonmail|mail|zoho|icloud|yandex|"
            + "?!gmail|hotmail|yahoo|outlook|aol|protonmail|mail|zoho|icloud|yandex)[a-zA-Z0-9.-]+)"
            + "\\.(com|net|org|info|biz|edu|gov|us|uk|de|fr|jp|es|cn|mx|es|gov|edu)");
    }
}
```

Figura 4.24: Validación del campo email.

Finalmente, en la clase de Usuario se debe llamar a la anotación @CustomEmail para que ese campo realice la validación del email que se ha diseñado. Además, es recomendable incluir siempre un mensaje para indicarle al usuario que ese campo no es correcto.

Otra forma de validar los formularios es como se indica en la figura 4.25.

```
@Column(name="email")
@CustomEmail(message = "El formato del email no es válido")
private String email;
```

Figura 4.25: Anotación CustomEmail.

También existen validaciones predeterminadas, como pueden ser @Digits para indicar el tamaño del campo precio o @DecimalMin para indicar que el precio debe ser un número positivo.

```
@Digits(integer = 10, fraction = 2, message = "El precio debe ser un número válido con máximo 10 dígitos enteros y 2 decimales")
@DecimalMin(value = "0.0", inclusive = false, message = "El precio debe ser un número positivo")
@Column(name="precio")
private Float precio;
```

Figura 4.26: Anotaciones predefinidas para números.

Capítulo 5

Conclusiones y trabajo futuro

5.1.- CONCLUSIONES

Finalmente, se han cubierto todos los objetivos planteados, así como algunos adicionales que surgieron durante el desarrollo del proyecto. En primer lugar, la gestión de reservas de productos para menús, personal de catering y fincas para eventos se ha desarrollado de manera completa. Los usuarios ahora pueden reservar estos servicios sin riesgo de solapamientos, ya que si una finca está reservada en un determinado periodo, no estará disponible para otras reservas. Esto facilitará enormemente la planificación y organización de eventos, evitando conflictos y mejorando la eficiencia en la gestión de recursos.

En segundo lugar, el desarrollo de la herramienta para personalizar menús a partir de los productos disponibles ahorrará mucho tiempo a los organizadores, permitiéndoles crear combinaciones de platos de manera rápida y sencilla. Esta funcionalidad no solo agiliza el

proceso de planificación, sino que también asegura que los menús se adapten mejor a las necesidades y preferencias de los clientes.

En tercer lugar, la plataforma permite la contratación de personal de catering de forma eficiente, proporcionando una lista de opciones y disponibilidad que facilita la selección del equipo adecuado para cada evento. Los organizadores pueden acceder a un listado detallado de perfiles y seleccionar al personal que mejor se ajuste a los requerimientos del evento, optimizando así la calidad del servicio.

Finalmente, la posibilidad de reservar fincas para eventos mediante la plataforma elimina la necesidad de múltiples llamadas y coordinaciones, centralizando todo el proceso en un solo lugar. Esto no solo simplifica la logística, sino que también proporciona una visión clara y actualizada de las disponibilidades y características de cada finca, permitiendo a los usuarios tomar decisiones informadas y adecuadas a sus necesidades específicas.

En conclusión, la implementación de este sistema de reservas integral mejora significativamente la eficiencia en la organización y gestión de eventos, proporcionando una herramienta robusta y fácil de usar para los organizadores. La centralización y automatización de estos procesos no solo ahorra tiempo y reduce errores, sino que también garantiza una experiencia más fluida y satisfactoria para todos los usuarios.

5.2.- POSIBLES DESARROLLOS FUTUROS

Para comenzar, se puede introducir un nuevo rol dentro del sistema, el rol de diseñador. Este rol estará específicamente dedicado a la configuración estética de la aplicación web, dar de alta imágenes en el carrousel, dar de alta reseñas, etc., permitiendo así liberar al administrador de la carga de trabajo asociada con el diseño y permitiéndole concentrarse en la gestión de reservas y usuarios.

Otra mejora significativa sería la incorporación de roles autónomos, los cuales ofrecen sus servicios o bienes de forma remunerada. Dentro de estos roles, tendríamos el rol del trabajador. Esta adición permitirá que los trabajadores se registren en la aplicación y ofrezcan sus servicios, ampliando así la diversidad de servicios disponibles para los usuarios y proporcionando una experiencia más completa y personalizada.

Por otro lado, se contempla la inclusión del rol de propietario, dirigido a aquellos usuarios que dispongan de propiedades, como fincas o espacios para eventos. Estos propietarios podrán dar de alta sus propiedades en la plataforma, permitiendo así que los clientes realicen reservas directamente sobre ellas, lo que ampliará la variedad de opciones disponibles y aumentará la versatilidad de la plataforma en su conjunto.

Una vez que se hayan establecido estos roles adicionales, se proporcionará información clara sobre qué personal y ubicaciones están vinculados directamente a la aplicación y cuáles son externos. Se comunicará a los usuarios que esta expansión podría implicar una ligera demora en el proceso de reserva debido a la participación de múltiples partes en su aceptación. Este enfoque transparente garantizará una comprensión clara por parte de los usuarios y una gestión eficiente de las expectativas.

Además, se planea la implementación de un sistema de notificaciones por correo electrónico. Este sistema se encargará de informar a los usuarios sobre cualquier modificación en el estado de sus reservas, asegurando una comunicación eficaz y transparente con los clientes y mejorando su experiencia de usuario en general. Por último un contador de unidades cada vez que se añade un artículo en el carrito.





Bibliografía

- [1] La importancia de una página web para tu restaurante
<https://www.bedoyahosteleria.es/pagina-web-restaurante/>
Editorial (año edición) / 03/03/2024

- [2] Spring Framework: Qué es y por qué usarlo
<https://ifgeekthen.everis.com/es/spring-framework>
Equipo Geek (07/11/2018) / 03/03/2024

- [3] Spring vs Spring Boot: Cual es la diferencia?
Pablo Miranda
<https://www.mirandas.work/es/posts/spring-vs-spring-boot-cual-es-la-diferencia/>
Mirandas (18/06/2019) / 03/03/2024

- [4] *Twitter Bootstrap Web Development* (1st edición). Packt Publishing. p. 100.
David Cochran
Packt Publishing(noviembre 2012) / 03/03/2021
- [5] Las 5 desafíos clave a los que se enfrenta la industria del catering
Paul Denley
<https://www.caterquip.co.uk/blog/the-5-key-challenges-facing-the-catering-industry>
Paul Denley (13/10/2019)/ 23/03/2024
- [6] Oportunidades y amenazas del mercado del catering
Modelos de plan de negocio
<https://modelosdeplandenegocios.com/oportunidades-amenazas-mercado-catering>
Editorial (año edición)/ 23/03/2024
- [7] Caterspot
<https://www.caterspot.sg/>
24/03/2024
- [8] La Seu Catering
<https://laseucatering.com/catering-eventos/>
04/04/2024
- [9] Catering Ya
<https://cateringya.com/>
13/04/2024
- [10] Spring Tools IDE
Programación móvil
<http://programandomovil.blogspot.com/2017/03/spring-tools-ide.html>
Gabriel CrAv/ 27/04/2024
- [11] Spring Tool Suite: el entorno de desarrollo definitivo
<https://wiki.yowu.dev/>
YOWU.DEV/ 27/04/2024
- [12] Spring Project
<https://spring.io/projects>
27/04/2024
- [14] Qué es WAMP (Guía para principiantes)
<https://www.hostinger.es/tutoriales/que-es-wamp/>
Gustavo B. / 27/04/2024

- [15] XAMPP
<https://www.ecured.cu/XAMPP>
EcuRed / 27/04/2024
- [16] Bringing MySQL to the web
<https://www.phpmyadmin.net/>
phpMyAdmin /27/04/2024
- [17] Dia
[https://es.wikipedia.org/wiki/Dia_\(programa\)](https://es.wikipedia.org/wiki/Dia_(programa))
27/04/2022 /28/04/2024
- [18] GanttProject
<https://www.ganttproject.biz/about>
GanttProject /18/05/2024
- [19] GanttProject
<https://www.ganttproject.biz/about>
Wikipedia/18/05/2024
- [20] Maven
<https://es.wikipedia.org/wiki/Maven>
Wikipedia/06/05/2024
- [21] Java:¿Qué es Maven?¿Qué es el archivo pom.xml?
<https://www.campusmvp.es/recursos/post/java-que-es-maven-que-es-el-archivo-pom-xml.aspx>
Jose Manuel Alarcón/06/05/2024
- [22] Diferencias entre librerías y framework
<https://www.mytaskpanel.com/diferencias-librerias-frameworks/>
My Task Panel consulting /08/05/2024
- [23] ¿Por qué utilizar un framework o librería web?
<https://gabrielchavez.me/por-que-utilizar-un-framework/>
Gabriel Chávez /08/05/2024
- [24] ¿Qué es Spring Framework?
<https://www.knowmadmood.com/blog/qu-es-spring-framework-caractersticas-i>
Javier Cuervas /08/05/2024

- [25] Lo que necesitas saber de Spring Boot
<https://barcochrist.medium.com/lo-que-necesitas-saber-de-spring-boot>
Christian Barco (02/08/2022) / 08/05/2024
- [26] Tipos de inyección de dependencias con Spring
<https://proitsolution.com.ve/inyeccion-de-dependencias-spring/>
Juan Moreno (06/12/2017) / 08/05/2024
- [27] Guía de anotaciones de Spring Framework
<https://mvinntec.com/mvinnovaciontecnologica/guia-de-anotaciones-de-spring-framework/>
Mvinntec (06/02/2020) / 08/05/2024
- [28] Bootstrap (framework)
[https://es.wikipedia.org/wiki/Bootstrap_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework))
Wikipedia(10/05/2024) / 11/05/2024
- [29] Java Platform, Standard Edition 8 API Specification.
<https://docs.oracle.com/javase/8/docs/api/>
Oracle / 11/05/2024
- [30] JavaServer Pages Standard Tag Library
https://es.wikipedia.org/wiki/JavaServer_Pages_Standard_Tag_Library
Wikipedia (21/03/2023) / 11/05/2024
- [31] JavaMail API documentation.
<https://javaee.github.io/javamail/docs/api/>
Oracle(2018) / 11/05/2024
- [32] MySQL Connector/J Developer Guide.
<https://dev.mysql.com/doc/connector-j/en/>
Oracle(10/05/2024) / 12/05/2024
- [33] El patrón modelo-vista-controlador: Arquitectura y framework explicados.
<https://www.freecodecamp.org/el-modelo-de-arquitectura-view-controller-pattern/>
Rafael D.Hernandez / 12/05/2024
- [34] HTML5.
<https://es.wikipedia.org/wiki/HTML5>
Wikipedia (06/05/2024) / 12/05/2024

- [35] Introducción al CSS: qué es, para qué sirve y 10 preguntas frecuentes .
<https://blog.hubspot.es/website/que-es-css>
Diego Santos(25/07/2023) / 12/05/2024
- [36] ¿Qué es JavaScript? .
https://developer.mozilla.org/es/Learn/JavaScript/First_steps/What_is_JavaScript
Mdn web docs(02/08/2023) / 12/05/2024
- [37] Qué es Java, para qué sirve, características e historia.
<https://blog.hubspot.es/website/que-es-java#caracteristicas>
Maria Coppola (13/01/2023) / 13/05/2024
- [38] ¿Qué es Java?
<https://aws.amazon.com/es/what-is/java/>
Amazon Web Services (2023) / 13/05/2024
- [39] Qué es JSP
<https://desarrolloweb.com/articulos/831.php>
Miguel Angel Alvarez (08/07/2002) / 13/05/2024
- [40] JavaServer Pages
https://es.wikipedia.org/wiki/JavaServer_Pages
Wikipedia (08/05/2024) / 13/05/2024
- [41] ¿Qué es SQL (lenguaje de consulta estructurada)
<https://aws.amazon.com/es/what-is/sql/>
Amazon Web Services (2023) / 13/05/2024
- [42] JavaServer Pages
<https://es.wikipedia.org/wiki/SQL>
Wikipedia (10/05/2024) / 13/05/2024
- [43] Ciclo de vida del Software todo lo que necesitas saber
<https://intelequia.com/es/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>
Intelequia(28/11/2020) / 13/05/2024
- [44] Apuntes de Ingeniería del software
Jesús Javier Rodríguez Sala
<http://umh2792.edu.umh.es/material/teoria/>
13/05/2024

- [45] El ciclo de vida de un sistema de información
Fernando Berzal
<https://elvex.ugr.es/idbis/db/docs/lifecycle.pdf>
13/05/2024



Anexo I

Casos de Uso

En este anexo se se van a describir en detalle los casos de uso de los tres roles de usuarios del proyecto, tal y como ya se indicó en el capítulo 4, la aplicación dispone de 3 tipos de usuario que presentan una relación de herencia tal y como se muestra en la figura AI.1.

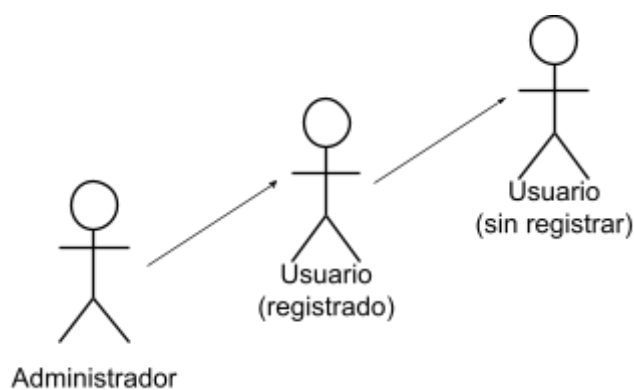


Figura AI.1: Diagrama jerárquico de los usuarios

AI.1.- USUARIO NO REGISTRADO

El usuario no registrado podrá registrarse o bien navegar por la aplicación viendo los productos disponibles, el personal, la finca, contactar con la empresa y ver su ubicación, pero sin poder reservar ninguno de los servicios que aparecen en ella.

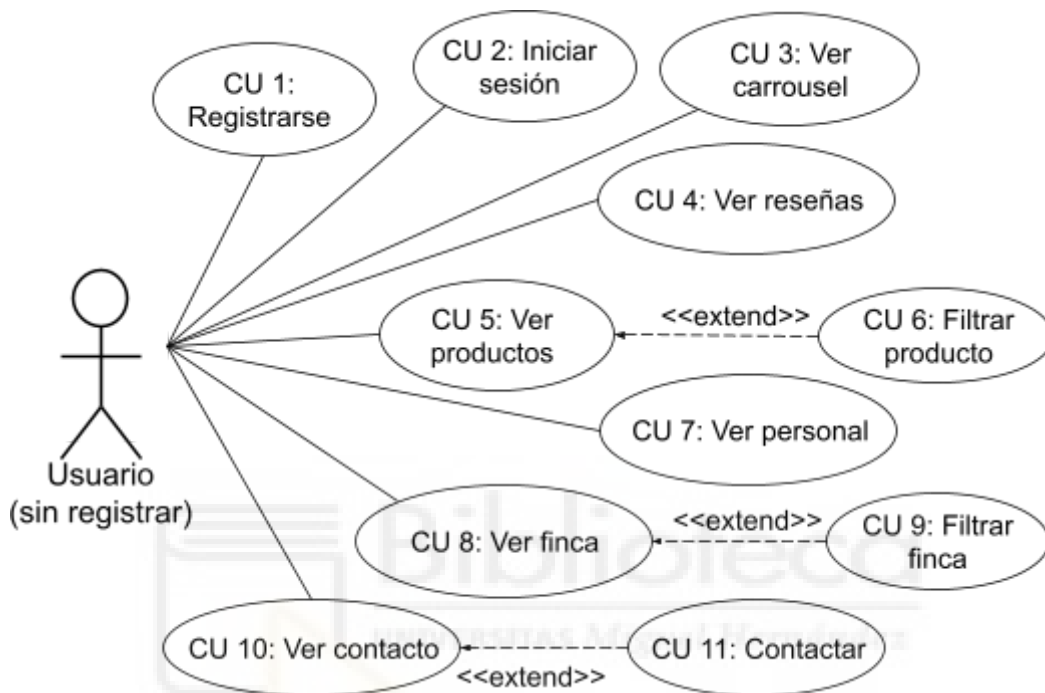


Figura AI.2: Diagrama casos de uso del usuario no registrado

Tabla AI.1: Caso de uso 1 - “Registrarse”

C.U. 1	<i>Registrarse</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Registrar un nuevo usuario en la aplicación.
Dependencias	Ninguna
Precondición	Estar en la página de inicio de sesión de la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario no registrado pulsa el icono de usuario en la página de inicio. 2. El sistema muestra un formulario de registro con campos obligatorios como nombre, dirección de correo electrónico y contraseña. 3. El usuario rellena el formulario de registro con la información requerida. 4. El usuario pulsa el botón "Registrarse". 5. El sistema valida los datos del formulario. Si se encuentra algún dato incorrecto o falta algún dato obligatorio, se muestra un mensaje de error y se ofrece al usuario la opción de corregirlo. 6. Si los datos del formulario son válidos, la aplicación registra al nuevo usuario. 7. La aplicación muestra la página de login.
Poscondición	La aplicación registra al nuevo usuario , después muestra la página de login para iniciar sesión.
Excepciones	<ul style="list-style-type: none"> • En el paso 5, se validan los datos del formulario, si alguno es incorrecto o falta algún dato obligatorio o si se ha duplicado el identificador del usuario se indicará con un mensaje de error dando al usuario la opción de subsanarlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal, se sale de la opción de “Registrarse”.

Tabla AI.2: Caso de uso 2 - “Iniciar sesión”

C.U. 2	<i>Iniciar sesión</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Permitir a los usuarios registrados iniciar sesión en la aplicación para acceder a sus privilegios y funcionalidades.
Dependencias	<i>C.U.1</i>
Precondición	El usuario debe estar registrado en la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario hace clic en el icono de usuario en la página de inicio. 2. El sistema muestra un formulario de inicio de sesión que requiere al usuario ingresar su nombre de usuario y contraseña. 3. El usuario completa el formulario de inicio de sesión con su información. 4. El usuario presiona el botón "Iniciar Sesión". 5. El sistema valida los datos del formulario. Si se encuentra algún dato incorrecto o falta algún dato obligatorio, se muestra un mensaje de error y se le da al usuario la opción de corregirlo. 6. Si los datos del formulario son válidos, la aplicación inicia la sesión del usuario y actualiza la aplicación para ofrecer los servicios de acuerdo a sus privilegios.
Poscondición	El usuario ha iniciado sesión con éxito y la aplicación se actualiza para proporcionar servicios y funcionalidades personalizados según los privilegios del usuario.
Excepciones	<ul style="list-style-type: none"> • En el paso 5, si se validan los datos del formulario y se encuentra un error, se muestra un mensaje de error al usuario y se le permite corregir la información.
Comentarios	El usuario que no esté registrado no podrá iniciar sesión.

Tabla AI.3: Caso de uso 3 - “Ver carrousel”

C.U. 3	<i>Ver Carrousel</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Permite a los usuarios visualizar un carrousel en la página principal de la aplicación. Las imágenes en el carrousel se desplazan automáticamente a lo largo de un intervalo de tiempo, lo que permite a los usuarios ver las distintas imágenes en la presentación.
Dependencias	<i>Ninguna</i>
Precondición	El usuario debe estar en la página de inicio de la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. Cuando el usuario accede a la página de inicio de la aplicación, el carrousel se muestra automáticamente. 2. El carrousel cambia de imagen automáticamente cada 3 segundos.
Poscondición	El usuario ha tenido la oportunidad de ver todas las imágenes en el carrousel a medida que se desplazan automáticamente.
Excepciones	Ninguna
Comentarios	El carrousel estará disponible para todos los usuarios, bien estén dados de alta o no y solo se verá en la página principal..

Tabla AI.4: Caso de uso 4 - “Ver reseñas clientes”

C.U. 4	<i>Ver Reseñas Clientes</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Quedar informado de las opiniones de los clientes que proporcionan sobre los productos o servicios ofrecidos por la empresa.
Dependencias	Ninguna
Precondición	El usuario debe estar en la página de inicio de la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. Cuando el usuario accede a la página de inicio de la aplicación, las reseñas de clientes se encuentran cerca del final de la página. 2. Para ver más reseñas, el usuario puede pulsar el icono de la flecha hacia la derecha o la izquierda para desplazar las reseñas hacia un lado u otro.
Poscondición	El usuario ha tenido la oportunidad de ver todas las reseñas de clientes disponibles en la página.
Excepciones	Ninguna
Comentarios	Las reseñas estarán disponibles para todos los usuarios, bien estén dados de alta o no y solo se verán en la página principal.

Tabla AI.5: Caso de uso 5 - “Ver productos”

C.U. 5	<i>Ver Productos</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Ver cada producto con su precio y su correspondiente descripción.
Dependencias	Ninguna
Precondición	El usuario debe estar en la página de productos de la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa la opción de productos en el menú principal 2. Se muestran todos los productos. Al pasar el ratón por encima de un producto se muestran sus detalles y un botón de “Añadir a carrito”. 3. El usuario puede aplicar filtros por categoría en los productos (CU6 extend).
Poscondición	El usuario ha tenido la oportunidad de ver todos los productos disponibles, incluyendo información sobre sus precios e ingredientes.
Excepciones	<ul style="list-style-type: none"> • En el paso 2 si pulsas añadir producto salta el login, ya que es una opción solo para usuarios registrados.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal, se sale de la página de productos.

Tabla AI.6: Caso de uso 6 - “Filtrar producto”

C.U. 6	<i>Filtrar Producto</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Permite realizar una búsqueda de uno o varios productos que cumplan una determinada condición.
Dependencias	<i>C.U.5</i>
Precondición	El usuario debe estar en la página de productos de la aplicación
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario puede explorar por categorías los diferentes productos pulsando el botón de cada categoría. 2. El usuario obtiene información sobre los productos por categoría.
Poscondición	El usuario queda informado sobre los productos disponibles en la base de datos que cumplen el criterio de búsqueda.
Excepciones	
Comentarios	El usuario puede pulsar en cualquier momento la opción “TODO” y la aplicación volverá a mostrar todos los productos sin filtro según corresponda. Si el usuario pulsa cualquier otra opción del menú principal se sale de la página de productos.

Tabla AI.7: Caso de uso 7 - “Ver personal”

C.U. 7	<i>Ver Personal</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Permitir a los usuarios ver el personal disponible en la aplicación
Dependencias	Ninguna
Precondición	El usuario debe estar en la página de personal de la aplicación
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de personal en la aplicación. 2. El usuario puede al pasar el ratón por encima de cada personal de contratación y se muestran sus detalles y un botón de “Contratar”. 3. El usuario obtiene información sobre el personal, incluyendo detalles sobre su profesión, disponibilidad y tarifas.
Poscondición	El usuario queda informado sobre el personal disponible en la aplicación y tiene acceso a información relevante para tomar decisiones sobre la contratación de servicios.
Excepciones	<ul style="list-style-type: none"> • En el paso 2 si pulsas contratar personal salta el login, ya que es una opción solo para usuarios registrados.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal, se sale de la página de personal.

Tabla AI.8: Caso de uso 8 - “Ver fincas”

C.U. 8	<i>Ver Fincas</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Permite a los usuarios explorar las fincas disponibles en la aplicación, las cuales pueden ser utilizadas para la organización de eventos. Los usuarios tienen la opción de seleccionar sobre diversas fincas disponibles, y además, pueden realizar búsquedas específicas por fecha para verificar la disponibilidad en un período particular.
Dependencias	Ninguna
Precondición	El usuario debe encontrarse en la página de fincas de la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de fincas en la aplicación. 2. El usuario puede explorar las diferentes fincas disponibles, obteniendo información detallada sobre cada una, incluyendo datos como ubicación, capacidad, servicios ofrecidos y tarifas. Además de tener la opción de reservar. 3. El usuario tiene la opción de realizar una búsqueda específica por fecha para verificar la disponibilidad de las fincas en un período particular. (CU9 extend)
Poscondición	El usuario queda informado sobre las opciones de fincas disponibles y, si así lo desea, puede verificar la disponibilidad de fincas para eventos en una fecha específica.
Excepciones	<ul style="list-style-type: none"> • En el paso 2 si pulsas reservar finca salta el login, ya que es una opción solo para usuarios registrados.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal, se sale de la página de fincas.

Tabla AI.9: Caso de uso 9 - “Filtrar finca”

C.U. 9	<i>Filtrar Finca</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Permitir a los usuarios hacer una búsqueda por fechas de las fincas que cumplan la condición.
Dependencias	<i>C.U.8</i>
Precondición	El usuario debe estar en la página de fincas de la aplicación
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de fincas en la aplicación. 2. El usuario selecciona la fecha para la que busca la finca 3. El usuario pulsa el botón “Filtrar”
Poscondición	El usuario queda informado sobre las fincas disponibles en la base de datos que cumplen el criterio de búsqueda
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de la página de finca.

Tabla AI.10: Caso de uso 10 - “Ver contacto”

C.U. 10	<i>Ver Contacto</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Permitir a los usuarios acceder a la página de contacto de la aplicación, donde pueden ver la ubicación de la empresa en un mapa y acceder a un formulario de contacto. Esto facilita la comunicación y la ubicación de la empresa.
Dependencias	Ninguna
Precondición	El usuario debe estar en la página de contacto de la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. Acceder a la página de contacto en la aplicación. 2. El usuario puede acceder al formulario de contacto para enviar un mensaje a la empresa (CU10 extend) 3. En la página de contacto, el usuario puede ver la ubicación de la empresa en un mapa.
Poscondición	El usuario puede enviar un mensaje o ver la ubicación de la empresa, lo que facilita la comunicación y su ubicación.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal, se sale de la opción de la página de contacto

Tabla AI.11: Caso de uso 11 - “Contactar”

C.U. 11	<i>Contactar</i>
Actores	Usuario no registrado, usuario registrado, administrador
Descripción	Este caso de uso permite a los usuarios, estando en la página de contacto de la aplicación, completar un formulario de contacto para enviar un mensaje solicitando la información que desean. Los usuarios pueden utilizar este formulario para comunicarse con la empresa.
Dependencias	<i>C.U.8</i>
Precondición	El usuario debe estar en la página de contacto de la aplicación
Secuencia normal	<ol style="list-style-type: none"> 1. La aplicación muestra un formulario con los datos de nombre, dirección de correo electrónico, asunto y mensaje. 2. El usuario rellena uno o alguno de los campos del formulario. 3. El usuario pulsa el botón "Enviar" para transmitir el mensaje a la empresa.
Poscondición	El usuario envía un mensaje a la empresa solicitando la información que desea.
Excepciones	<ul style="list-style-type: none"> • En el paso 2, se validan los datos del formulario. Si se encuentra algún dato incorrecto o falta algún dato obligatorio, se muestra un mensaje de error y se le da al usuario la opción de corregirlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal, se sale de la opción de “Contactar”.

Tabla AI.12: Caso de uso 12 - “Cerrar sesión”

C.U. 12	<i>Cerrar sesión</i>
Actores	Usuario registrado, administrador
Descripción	Permite a un usuario registrado o a un administrador cerrar sesión en la webapp. Cuando se cierra la sesión, el usuario ya no tiene acceso a las funcionalidades protegidas y se desconecta de su cuenta.
Dependencias	<i>C.U.2</i>
Precondición	El usuario debe tener una sesión iniciada en la webapp.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario registrado o el administrador va al icono de usuario en la webapp. 2. Luego, en el desplegable pulsa el botón "Cerrar Sesión".
Poscondición	El usuario registrado o el administrador deja de estar conectado y se cierra la sesión. La aplicación vuelve a la página principal.
Excepciones	Ninguna
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal, se sale de la opción de “Cerrar sesión”.

AI.2.- USUARIO REGISTRADO

El usuario registrado es aquel que dispone de privilegios de reserva, con lo cual iniciando sesión podrá reservar cualquiera de los servicios ofrecidos.

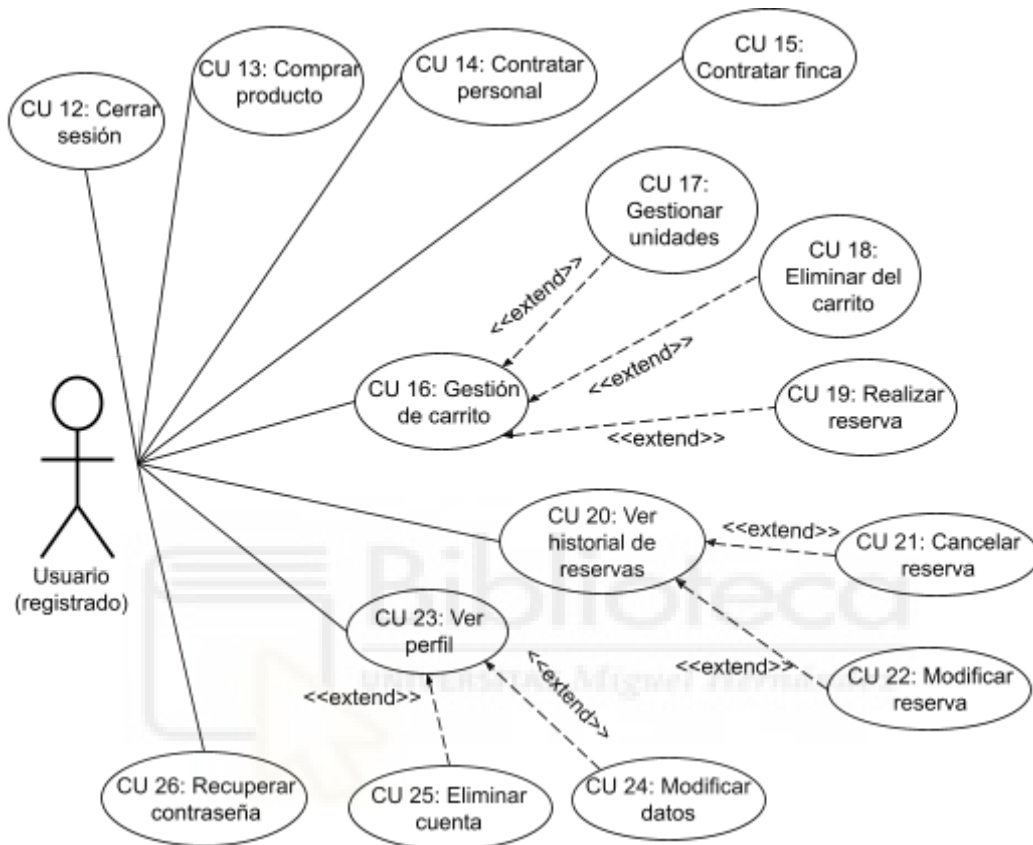


Figura AI.3: Diagrama casos de uso del usuario registrado

Tabla AI.13: Caso de uso 13 - “Comprar producto”

C.U. 13	<i>Comprar producto</i>
Actores	Usuario registrado, administrador
Descripción	Permitir a un usuario registrado o administrador comprar un producto en la aplicación. Al pasar el cursor sobre el producto deseado en la página de producto, aparecerá la opción de añadirlo a la cesta. Al seleccionar esta opción, el producto se añadirá al carrito de la reserva para su compra.
Dependencias	<i>C.U.2</i>
Precondición	El usuario debe haber iniciado sesión en la webapp.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. En el menú principal, selecciona la pestaña de productos. 3. El usuario pasa el cursor sobre el producto deseado en la página de producto. 4. Se muestra la opción de añadir el producto a la cesta. 5. El usuario selecciona la opción de añadir a la cesta.
Poscondición	El producto deseado se agrega al carrito de la reserva, donde se pueden gestionar las unidades del producto.
Excepciones	Si el usuario no está registrado lo redirige a la página de login.
Comentarios	<p>Cuando se añade un producto al carrito si es el primer elemento se da de alta un nuevo evento para el usuario.</p> <p>Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones comprar producto.</p>

Tabla AI.14: Caso de uso 14 - “Contratar personal”

C.U. 14	<i>Contratar personal</i>
Actores	Usuario registrado, administrador
Descripción	Estando en la página de personal de la aplicación, en la última columna de la tabla tenemos la opción de contratar. Debemos pulsar ese botón para contratar el servicio que ofrece la persona.
Dependencias	<i>C.U.2</i>
Precondición	El usuario debe haber iniciado sesión en la webapp.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. En el menú principal, navega hacia la pestaña de personal. 3. El usuario selecciona la persona cuyo servicio desea contratar. 4. En la última columna de la tabla, se encuentra la opción de contratar. 5. El usuario pulsa la opción de contratar.
Poscondición	Obtener en el carrito las contrataciones de personal deseadas
Excepciones	Si el usuario no está registrado lo redirige a la página de login.
Comentarios	<p>Cuando se añade una contratación al carrito si es el primer elemento se da de alta un nuevo evento para el usuario.</p> <p>Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones contratar personal.</p>

Tabla AI.15: Caso de uso 15 - “Contratar finca”

C.U. 15	<i>Contratar finca</i>
Actores	Usuario registrado, administrador
Descripción	Estando en la página de finca de la aplicación, tenemos la opción de reservar. Debemos pulsar ese botón para reservar esa finca
Dependencias	<i>C.U.2</i>
Precondición	Tener iniciada sesión en la webapp
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. En el menú principal, navega hacia la pestaña de finca. 3. Agrega la finca deseada al carrito de la reserva.
Poscondición	Obtener en el evento un lugar para su realización
Excepciones	Si el usuario no está registrado lo redirige a la página de login.
Comentarios	Solo se puede reservar una finca por evento. Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones contratar finca.



Tabla AI.16: Caso de uso 16 - “Gestión de carrito”

C.U. 16	<i>Gestión de carrito</i>
Actores	Usuario registrado, administrador
Descripción	Estando en la página del carrito de la aplicación, podemos gestionar nuestra reserva.
Dependencias	<i>C.U.2,C.U.11,C.U.12,C.U.13</i>
Precondición	El usuario debe haber iniciado sesión en la webapp y deben existir elementos en el carrito.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. En el menú principal, selecciona la opción deseada para realizar la reserva, ya sea productos, fincas o personal. 3. Agrega los elementos deseados al carrito de reserva. 4. En el menú principal, navega hacia la pestaña de carrito 5. Desde el carrito puedes gestionar las unidades o cantidades de los productos, eliminar elementos no deseados y realizar la reserva confirmándola.
Poscondición	Tener los elementos deseados en el carrito y así realizar la reserva.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones gestionar carrito.

Tabla AI.17: Caso de uso 17 - “Gestionar unidades”

C.U. 17	<i>Gestionar unidades</i>
Actores	Usuario registrado, administrador
Descripción	Permitir a los usuarios gestionar las unidades de los productos que han agregado a su carrito de compras en la webapp. Pueden aumentar o disminuir la cantidad de unidades de los productos según sus necesidades.
Dependencias	<i>C.U.2, C.U.16</i>
Precondición	El usuario debe haber iniciado sesión en la webapp y deben existir productos en el carrito
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario inicia sesión en la webapp. 2. El usuario agrega productos a su carrito de compras 3. El usuario navega hacia su carrito de compras. 4. En la página del carrito, el usuario puede aumentar o disminuir la cantidad de unidades de los productos pulsando el botón “+” o “-”. 5. El sistema actualiza la cantidad de unidades de cada producto en el carrito.
Poscondición	El usuario obtiene la cantidad exacta de unidades de cada producto necesarias para su evento.
Excepciones	Solo esta disponible este caso de uso para los productos
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones gestionar unidades.

Tabla AI.18: Caso de uso 18 - “Contratar finca”

C.U. 18	<i>Eliminar del carrito</i>
Actores	Usuario registrado, administrador
Descripción	Permitir eliminar productos, personal o fincas que hayan sido previamente añadidos al carrito para la reserva de un evento en la webapp. Los elementos seleccionados se quitan del carrito, lo que permite al usuario ajustar su reserva según sus necesidades.
Dependencias	<i>C.U.2, C.U.16</i>
Precondición	El usuario debe haber iniciado sesión en la webapp y deben existir elementos en el carrito que el usuario desee eliminar.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario navega al carrito de reserva del evento, donde se encuentran los elementos que ha agregado previamente. 2. El usuario selecciona el elemento que desea eliminar. 3. El usuario confirma la eliminación del elemento seleccionado. 4. El sistema actualiza el carrito, eliminando los elementos seleccionados.
Poscondición	Los elementos que el usuario no desea o no necesita se eliminan de su reserva de evento en el carrito.
Excepciones	
Comentarios	

Tabla AI.19: Caso de uso 19 - “Realizar reserva”

C.U. 19	<i>Realizar reserva</i>
Actores	Usuario registrado, administrador
Descripción	Permitir a un usuario registrado o un administrador realizar una reserva en la webapp. La reserva puede incluir productos, fincas o servicios de personal para eventos, y se gestiona a través del proceso de agregar elementos al carrito y finalizar la reserva.
Dependencias	<i>C.U.2, C.U.16</i>
Precondición	El usuario debe haber iniciado sesión en la webapp y deben existir elementos en el carrito que el usuario desee para su evento.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. El usuario navega al carrito de reserva del evento, donde se encuentran los elementos que ha agregado previamente. 3. Una vez esta todo como lo desea pulsa el botón reservar 4. Al usuario le llegará un correo indicando que tiene realizada la reserva a la espera de que se le confirme.
Poscondición	Tener los productos, o contrataciones o lugar para tu evento
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones del carrito para realizar la reserva.

Tabla AI.20: Caso de uso 20 - “Ver historial de reservas”

C.U. 20	<i>Ver historial de reservas</i>
Actores	Usuario registrado, administrador
Descripción	Permitir al usuario ver todas sus reservas realizadas
Dependencias	<i>C.U.2</i>
Precondición	El usuario debe haber iniciado sesión en la webapp y haber realizado alguna reserva anteriormente.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp 2. El usuario navega el icono de usuario, donde al poner el mouse encima sale un desplegable con varias opciones 3. Selecciona la opción de mis reservas
Poscondición	Se muestran las reservas realizadas
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de ver la reserva.

Tabla AI.21: Caso de uso 21 - “Cancelar reserva”

C.U. 21	<i>Cancelar reserva</i>
Actores	Usuario registrado, administrador
Descripción	Cancelar una reserva por la posible cancelación del evento
Dependencias	<i>C.U.2, C.U.20</i>
Precondición	El usuario debe haber iniciado sesión en la webapp y tener reservas realizadas
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. El usuario navega el icono de usuario, donde al poner el mouse encima sale un desplegable con la opción de mis reservas. 3. Dentro de mis reservas tienes que seleccionar la reserva que quieres ver. 4. Dentro de la reserva puedes pulsar el botón y cancelar la reserva.
Poscondición	La reserva queda cancelada.
Excepciones	Si la reserva ya está confirmada no podrá realizar esta acción.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de gestión de la reserva.

Tabla AI.22: Caso de uso 22 - “Modificar reservas”

C.U. 22	<i>Modificar reservas</i>
Actores	Usuario registrado, administrador
Descripción	Estando en la página de inicio de la aplicación, si bajamos casi hasta el final podremos leer las reseñas de los clientes.
Dependencias	<i>C.U.2, C.U.20</i>
Precondición	El usuario debe haber iniciado sesión en la webapp y tener reservas realizadas
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. El usuario navega el icono de usuario, donde al poner el mouse encima sale un desplegable con la opción de mis reservas 3. Dentro de mis reservas tienes que seleccionar la reserva que quieres ver. 4. Dentro de la reserva puedes pulsar el botón y modificar la reserva. Esto hace que se muevan los artículos al carrito de nuevo para su gestión. 5. Ir al carrito, configurar de nuevo la reserva y realizarla
Poscondición	Modificar una reserva realizada
Excepciones	
Comentario	En caso de tener elementos en el carrito, al modificar una reserva esos elementos son eliminados para realizar correctamente la gestión de la reserva. Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de gestión de la reserva.

Tabla AI.23: Caso de uso 23 - “Ver perfil”

C.U. 23	<i>Ver perfil</i>
Actores	Usuario registrado, administrador
Descripción	Permitir ver al usuario los datos de su perfil
Dependencias	<i>C.U.2</i>
Precondición	El usuario debe haber iniciado sesión en la webapp
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. El usuario, al poner el mouse sobre el icono de usuario sale un desplegable con la opción de mi perfil. 3. Pulsa la opción de mi perfil.
Poscondición	Ver los datos personales del usuario.
Excepciones	Esta opción solo está disponible para usuarios registrados.
Comentario	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones del perfil.

Tabla AI.24: Caso de uso 24 - “Modificar datos”

C.U. 24	<i>Modificar datos</i>
Actores	Usuario registrado, administrador
Descripción	Permitir modificar los datos del perfil del usuario.
Dependencias	<i>C.U.2, C.U.23</i>
Precondición	Estar registrado en la aplicación
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. El usuario, al poner el mouse sobre el icono de usuario sale un desplegable con la opción de mi perfil 3. Dentro de mi perfil, modifica los atributos que desee 4. Una vez modificada la información que desee debe pulsar el botón “guardar”
Poscondición	Modificar los datos personales del usuario
Excepciones	
Comentarios	El usuario que no esté registrado no podrá ver su perfil ya que no ha sido dado de alta previamente.

Tabla AI.25: Caso de uso 25 - “Eliminar cuenta”

C.U. 25	<i>Eliminar cuenta</i>
Actores	Usuario registrado, administrador
Descripción	Permitir darse de baja de la webapp.
Dependencias	<i>C.U.2, C.U.23</i>
Precondición	Estar registrado en la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. El usuario navega el icono de usuario, donde al poner el mouse encima sale un desplegable con la opción de mi perfil. 3. Una vez aquí debe pulsar el botón eliminar para borrar su cuenta.
Poscondición	Eliminar la cuenta y tus datos de la webapp
Excepciones	
Comentarios	El usuario que no esté registrado no podrá eliminar su perfil ya que no ha sido dado de alta previamente.



Tabla AI.26: Caso de uso 26 - “Recuperar contraseña”

C.U. 26	<i>Recuperar contraseña</i>
Actores	Usuario registrado, administrador
Descripción	Permitir a los usuarios registrados recuperar su contraseña de inicio de sesión en la webapp en caso de que la olviden.
Dependencias	<i>C.U.1</i>
Precondición	El usuario debe estar registrado en la webapp.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. El usuario navega el icono de usuario 3. El usuario intenta loguear a la webapp. 4. Si no lo consigue, tiene la opción de recuperar la contraseña pulsando “¿Olvidaste tu contraseña?” 5. El sistema envía un correo electrónico al usuario registrado con la contraseña. 6. Después de recuperar la contraseña, el usuario puede iniciar sesión correctamente en la webapp.
Poscondición	El usuario recupera su contraseña y puede iniciar sesión con éxito en la webapp.
Excepciones	<ul style="list-style-type: none"> • Si el correo electrónico del usuario no está registrado en la base de datos, se le notifica al usuario que la dirección de correo electrónico no existe en el sistema.
Comentarios	El objetivo es proporcionar a los usuarios una forma segura y conveniente de recuperar el acceso a sus cuentas en caso de olvidar su contraseña.

AI.3.- ADMINISTRADOR

El usuario con rol de administrador será el que gestione toda la aplicación. Podrá añadir, modificar o eliminar, imágenes del carrusel, reseñas de clientes, modificar cualquier producto, personal o finca.. En cuanto a la gestión de usuario el administrador podrá dar de alta, modificar o eliminar usuarios de su aplicación.

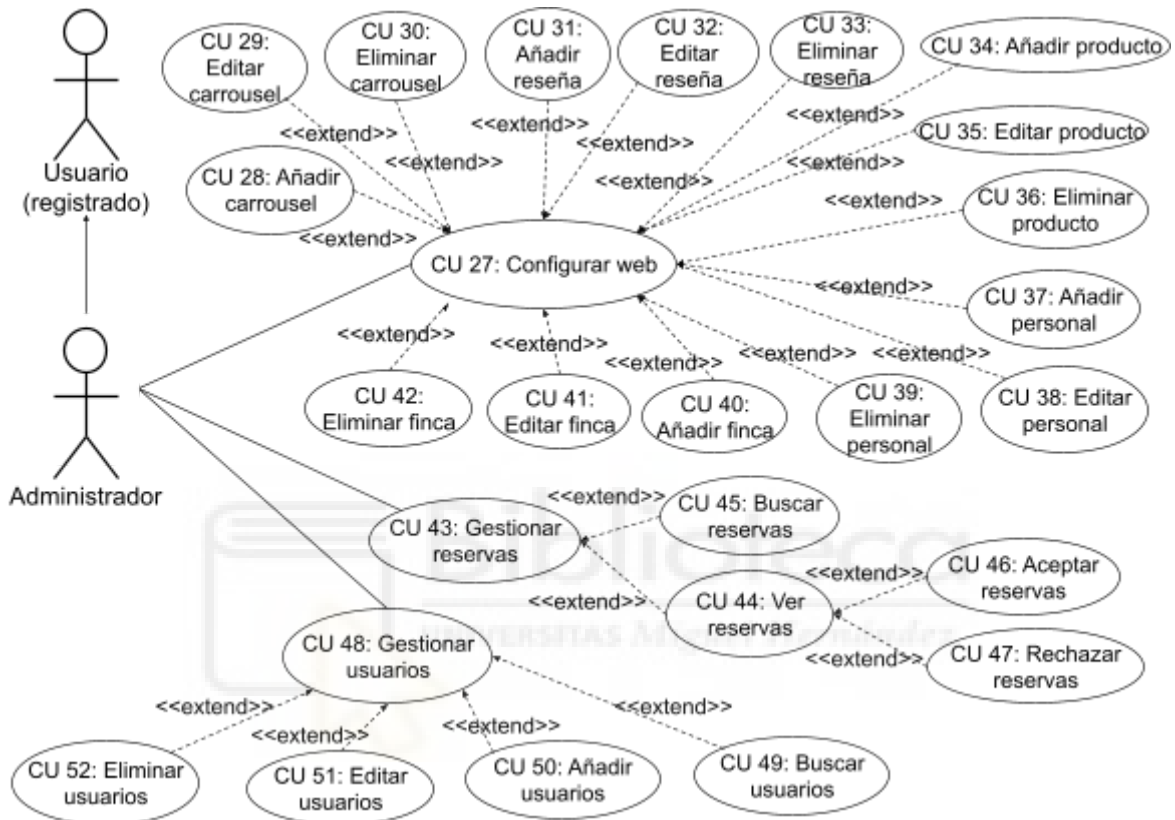


Figura AI.4: Diagrama casos de uso del administrador

Tabla AI.27: Caso de uso 27 - “Configurar web”

C.U. 27	<i>Configurar web</i>
Actores	Administrador
Descripción	Permitir personalizar la webapp ,pudiendo modificar y actualizar diversos elementos visuales y de contenido. El Administrador puede agregar y gestionar imágenes para el carrusel principal, incluir reseñas de clientes para destacar la satisfacción de los usuarios, agregar y gestionar productos, agregar y gestionar al personal de la empresa y sus roles, así como agregar y gestionar los detalles sobre las fincas disponibles.
Dependencias	<i>C.U.2</i>
Precondición	Tener iniciada sesión con el rol de administrador
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario accede a la webapp. 2. El usuario navega el icono de usuario y se loguea, si tiene asignado el rol de administrador se muestran las pestañas correspondientes.. 3. Navega hacia el en el panel de administración de la web. 4. Dentro de la configuración, elige la sección que desea personalizar (carrusel, reseñas, productos, personal, fincas). 5. Para cada sección seleccionada puede realizar diferentes acciones: <ul style="list-style-type: none"> - Carrousel: Añadir , editar o eliminar (CU28 - CU30 extend) carrousel. - Reseñas: Añadir, editar o eliminar (CU31 - CU33 extend) reseñas de clientes. - Productos: Añadir (CU34 extend), editar (CU35 extend) o eliminar (CU36 extend) productos. -Personal: Añadir, editar o eliminar (CU37 - CU39 extend)información del personal y sus roles. -Fincas: Añadir, editar o eliminar (CU40 - CU42 extend) detalles sobre las fincas disponibles. 6. El Administrador guarda los cambios realizados.
Poscondición	La aplicación muestra las opciones de alta, baja y modificación de los distintos menús principales..
Excepciones	
Comentarios	Si el administrador pulsa cualquier otra opción del menú principal se sale de las opciones de configuración de web.

Tabla AI.28: Caso de uso 28 - “Añadir carrousel”

C.U. 28	<i>Añadir carrousel</i>
Actores	Administrador
Descripción	Registrar un nuevo carrousel en la página de inicio de la aplicación.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de inicio. 3. Selecciona el botón "Nuevo" en la sección de carrousel para añadir una nueva imagen al carrusel. 4. Rellena el formulario ingresando el título y el subtítulo de la imagen, así como la imagen misma. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Se muestra una nueva imagen en el carrousel de imágenes de la página de inicio.
Excepciones	<ul style="list-style-type: none"> • En P4, se validan los datos del formulario, si alguno es incorrecto o falta algún dato obligatorio se indicará con un mensaje de error dando al usuario la opción de subsanarlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opción de añadir carrousel.

Tabla AI.29: Caso de uso 29 - “Editar carrousel”

C.U. 29	<i>Editar carrousel</i>
Actores	Administrador
Descripción	Permitir modificar el título, subtítulo o la imagen de un carrousel ya existente.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de inicio. 3. Selecciona el botón "Editar" en la sección de carrousel. 4. Modifica el título ,el subtítulo o la imagen misma. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Los cambios realizados en el título, subtítulo o imagen del carrousel se reflejan correctamente en la página de inicio.
Excepciones	<ul style="list-style-type: none"> ● En P4, se validan los datos del formulario, si alguno es incorrecto o falta algún dato obligatorio se indicará con un mensaje de error dando al usuario la opción de subsanarlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de editar carrousel.

Tabla AI.30: Caso de uso 30 - “Eliminar carousel”

C.U. 30	<i>Eliminar carousel</i>
Actores	Administrador
Descripción	Eliminar un carousel en la página de inicio de la aplicación.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de inicio. 3. Selecciona el botón "Eliminar" en la sección de carrusel para eliminar el carrusel. 4. Saldrá un pop up preguntando si estás seguro de eliminar el carousel. 5. Confirma la acción haciendo clic en el botón "Si".
Poscondición	Se elimina el carousel de la página de inicio y de la base de datos.
Excepciones	
Comentarios	Como medida de seguridad se agrega una confirmación antes de eliminar el carousel, para evitar eliminaciones accidentales. Si pulsa cualquier opción del menú superior te sales de las opciones de configuración de la web.

Tabla AI.31: Caso de uso 31 - “Añadir reseña”

C.U. 31	<i>Añadir reseña</i>
Actores	Administrador
Descripción	Registrar una reseña de un cliente en la página de inicio de la aplicación.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de inicio. 3. Selecciona el botón "Nuevo" en la sección de reseñas para añadir una nueva reseña 4. Rellena el formulario ingresando el título y el subtítulo de la imagen, así como la imagen misma. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Se muestra una nueva reseña de una cliente en la página de inicio.
Excepciones	<ul style="list-style-type: none"> • En P4, se validan los datos del formulario, si alguno es incorrecto o falta algún dato obligatorio se indicará con un mensaje de error dando al usuario la opción de subsanarlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de la opción de añadir reseña.

Tabla AI.32: Caso de uso 32 - “Editar reseña”

C.U. 32	<i>Editar reseña</i>
Actores	Administrador
Descripción	Permitir modificar el título, subtítulo o la imagen de una reseña ya existente.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de reseñas. 3. Selecciona el botón "Edit" en la sección de reseñas para modificarla reseña. 4. Modifica el título ,el subtítulo o la imagen misma. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Los cambios realizados en el título, subtítulo o imagen de la reseña se reflejan correctamente en la página de inicio.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de la opción de editar reseña.

Tabla AI.33: Caso de uso 33 - “Eliminar reseña”

C.U. 33	<i>Eliminar reseña</i>
Actores	Administrador
Descripción	Permite a un administrador eliminar una reseña de la página de inicio de la webapp..
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de inicio. 3. Selecciona el botón "Eliminar" en la sección de reseñas para eliminar la reseña 4. Saldrá un pop up preguntando si estás seguro de eliminar la reseña. 5. Confirma la acción haciendo clic en el botón "Si".
Poscondición	La reseña seleccionada se elimina de la página de inicio.
Excepciones	
Comentarios	Como medida de seguridad se agrega una confirmación antes de eliminar la reseña, para evitar eliminaciones accidentales. Si se pulsa cualquier opción del menú superior te sales de las opciones de configuración de la web.

Tabla AI.34: Caso de uso 34 - “Añadir producto”

C.U. 34	<i>Añadir producto</i>
Actores	Administrador
Descripción	Registrar un nuevo producto en la página de productos.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de productos. 3. Selecciona el botón "Nuevo" para añadir un nuevo producto 4. Ingresa el nombre, descripción, categoría, precio y la imagen del producto. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Se muestra un nuevo producto en la página de productos.
Excepciones	<ul style="list-style-type: none"> • En P4, se validan los datos del formulario, si alguno es incorrecto o falta algún dato obligatorio se indicará con un mensaje de error dando al usuario la opción de subsanarlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de la opción de dar de alta un producto.

Tabla AI.35: Caso de uso 35 - “Editar producto”

C.U. 35	<i>Editar producto</i>
Actores	Administrador
Descripción	Permite al administrador modificar los atributos de un producto en la página de productos.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de productos. 3. Selecciona el botón "Editar" para modificar un producto 4. Modifica el nombre, descripción, categoría, precio, unidades o la imagen del producto. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Los cambios realizados en el nombre, descripción, categoría, precio, unidades o la imagen del producto se reflejan correctamente en la página de productos.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de la opción de editar producto.

Tabla AI.36: Caso de uso 36 - “Eliminar producto”

C.U. 36	<i>Eliminar producto</i>
Actores	Administrador
Descripción	Permite al administrador eliminar un producto en la página de productos.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la sección de configuración de productos. 3. Selecciona el botón "Eliminar" en el producto que deseas eliminar. Saldrá un pop up preguntando si estás seguro de eliminar ese producto. 4. Confirma la acción haciendo clic en el botón "Si".
Poscondición	El producto seleccionado se elimina de la página de productos.
Excepciones	
Comentarios	Como medida de seguridad se agrega una confirmación antes de eliminar el producto, para evitar eliminaciones accidentales. Si pulsas cualquier opción del menú superior te sales de las opciones de configuración de la web.

Tabla AI.37: Caso de uso 37 - “Añadir personal”

C.U. 37	<i>Añadir personal</i>
Actores	Administrador
Descripción	Registrar un nuevo empleado en la página de personal.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de personal. 3. Selecciona el botón "Nuevo" para añadir un nuevo empleado. 4. Ingresa la profesión, precio por hora, horas y la imagen de la profesión. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Se muestra un nuevo empleado en la página de personal.
Excepciones	<ul style="list-style-type: none"> • En P4, se validan los datos del formulario, si alguno es incorrecto o falta algún dato obligatorio se indicará con un mensaje de error dando al usuario la opción de subsanarlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de la opción de añadir personal.

Tabla AI.38: Caso de uso 38 - "Editar personal"

C.U. 38	<i>Editar personal</i>
Actores	Administrador
Descripción	Permite al administrador modificar los atributos de un empleado en la página de personal.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de personal. 3. Selecciona el botón "Editar" para modificar un empleado 4. Modifica la profesión, experiencia, precio por hora, horas y la imagen de la persona. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Los cambios realizados en la profesión, experiencia, precio por hora, horas o la imagen de la profesion.se reflejan correctamente en la página de personal.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de la opción de editar personal.

Tabla AI.39: Caso de uso 39 - “Eliminar personal”

C.U. 39	<i>Eliminar personal</i>
Actores	Administrador
Descripción	Permite al administrador eliminar un empleado en la página de personal.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la sección de configuración de personal. 3. Selecciona el botón "Eliminar" en el empleado que deseas eliminar. Saldrá un pop up preguntando si estás seguro de eliminar ese empleado. 4. Confirma la acción haciendo clic en el botón "Si".
Poscondición	El personal seleccionado se elimina de la página de personal.
Excepciones	
Comentarios	Como medida de seguridad se agrega una confirmación antes de eliminar el personal, para evitar eliminaciones accidentales. Si pulsas cualquier opción del menú superior te sales de las opciones de configuración de la web.

Tabla AI.40: Caso de uso 40 - “Añadir finca”

C.U. 40	<i>Añadir finca</i>
Actores	Administrador
Descripción	Registrar una nueva finca en la página de fincas.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de finca. 3. Selecciona el botón "Nuevo" para añadir una nueva finca 4. Ingresa el nombre,dirección, características, fecha de disponibilidad, el precio y las imágenes de la finca. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Se muestra una nueva finca en la página de finca.
Excepciones	<ul style="list-style-type: none"> • En P4, se validan los datos del formulario, si alguno es incorrecto o falta algún dato obligatorio se indicará con un mensaje de error dando al usuario la opción de subsanarlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de la opción de añadir finca.

Tabla AI.41: Caso de uso 41 - "Editar finca"

C.U. 41	<i>Editar finca</i>
Actores	Administrador
Descripción	Permite al administrador modificar los atributos de una finca en la página de finca.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la opción configuración de finca. 3. Selecciona el botón "Editar" para modificar una finca. 4. Modifica el nombre,dirección, características, fecha de disponibilidad, el precio y las imágenes de la finca.. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Los cambios realizados en el nombre,dirección, características, fecha de disponibilidad, el precio o las imágenes de la finca.se reflejan correctamente en la página de finca.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de la opción de editar finca.

Tabla AI.42: Caso de uso 42 - "Eliminar finca"

C.U. 42	<i>Eliminar finca</i>
Actores	Administrador
Descripción	Permite al administrador eliminar una finca en la página de fincas.
Dependencias	<i>C.U.2, C.U.27</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de configuración, y elige la busca la sección de configuración de finca. 3. Selecciona el botón "Eliminar" en la finca que deseas eliminar. Saldrá un pop up preguntando si estás seguro de eliminar esa finca. 4. Confirma la acción haciendo clic en el botón "Si".
Poscondición	La finca seleccionada se elimina de la página de finca.
Excepciones	
Comentarios	Como medida de seguridad se agrega una confirmación antes de eliminar la finca, para evitar eliminaciones accidentales. Si pulsas cualquier opción del menú superior te sales de las opciones de configuración de la web.

Tabla AI.43: Caso de uso 43 - “Gestionar reserva”

C.U. 43	<i>Gestionar reservar</i>
Actores	Administrador
Descripción	Permite al administrador gestionar las reservas de la aplicación.
Dependencias	<i>C.U.2</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de reservas.
Poscondición	La aplicación muestra las opciones de búsqueda y de ver reserva.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de gestión de usuarios.



Tabla AI.44: Caso de uso 44 - “Ver reservas”

C.U. 44	<i>Ver reservas</i>
Actores	Administrador
Descripción	Permite al administrador ver todas las reservas realizadas en la webapp.
Dependencias	<i>C.U.2, C.U.43</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de reservas. 3. El administrador pulsando el botón ver reserva puede ver la reserva y aceptarla (CU 46 extend) o rechazarla (CU 47 extend)
Poscondición	El administrador de la reserva para su posterior gestión.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de reserva.

Tabla AI.45: Caso de uso 45 - “Buscar reservas”

C.U. 45	<i>Buscar reservas</i>
Actores	Administrador
Descripción	Permite al administrador buscar una reserva por su ID reserva, el nick del usuario o la fecha de la misma.
Dependencias	<i>C.U.2, C.U.43</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de reservas. 3. El administrador rellena alguno de los campos de búsqueda ID reserva, nick o fecha 4. Pulsa el botón filtrar del lado derecho del campo relleno.
Poscondición	El administrador queda informado sobre la o las reservas disponibles en la base de datos que cumplen el criterio de búsqueda.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de reserva.

Tabla AI.46: Caso de uso 46 - “Aceptar reservas”

C.U. 46	<i>Aceptar reservas</i>
Actores	Administrador
Descripción	Permite al administrador aceptar una reserva
Dependencias	<i>C.U.2, C.U.44</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de reservas. 3. El administrador pulsando el botón ver reserva . 4. A continuación dentro de la reserva pulsa el botón confirmar.
Poscondición	La reserva queda aceptada. El usuario recibe un correo con esta información.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de reserva.

Tabla AI.47: Caso de uso 47 - “Rechazar reservas”

C.U. 47	<i>Rechazar reservas</i>
Actores	Administrador
Descripción	Permite al administrador aceptar una reserva
Dependencias	<i>C.U.2, C.U.25</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de reservas. 3. El administrador pulsando el botón ver reserva . 4. A continuación dentro de la reserva pulsa el botón rechazar.
Poscondición	La reserva queda rechazada.El usuario recibe un correo con esta información.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de reserva.

Tabla AI.48: Caso de uso 48 - “Gestionar usuarios”

C.U. 48	<i>Gestionar usuarios</i>
Actores	Administrador
Descripción	Permite al administrador gestionar los usuarios de la aplicación.
Dependencias	<i>C.U.2</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de usuarios.
Poscondición	La aplicación muestra las opciones de búsqueda, alta, baja y modificación de usuarios.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de gestión de usuarios.



Tabla AI.49: Caso de uso 49 - “Buscar usuarios”

C.U. 49	<i>Buscar usuarios</i>
Actores	Administrador
Descripción	Permite al administrador buscar un usuario por su nick o por su nombre.
Dependencias	<i>C.U.2, C.U.48</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de usuarios. 3. Selecciona el campo por el que desea filtrar. Indica el valor a buscar y pulsa el botón "Filtrar" para buscar el usuario que deseas.
Poscondición	El administrador obtiene el registro del usuario que busca.
Excepciones	El administrador no encuentra al usuario buscado, salta un pop up indicando que el usuario no ha sido encontrado.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de usuario..

Tabla AI.50: Caso de uso 50 - “Añadir usuarios”

C.U. 50	<i>Añadir usuarios</i>
Actores	Administrador
Descripción	Registrar un nuevo usuario en la aplicación.
Dependencias	<i>C.U.2, C.U.48</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de usuarios 3. Selecciona el botón "Nuevo" para añadir un nuevo usuario. 4. Ingresa el nick,password,nombre,apellidos, email y el perfil del usuario.. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Se registra un nuevo usuario en la webapp con el perfil deseado, después la página se redirige al login.
Excepciones	<ul style="list-style-type: none"> • En P4, se validan los datos del formulario, si alguno es incorrecto o falta algún dato obligatorio o si se ha duplicado el identificador del usuario se indicará con un mensaje de error dando al usuario la opción de subsanarlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de gestión de usuarios

Tabla AI.51: Caso de uso 51 - "Editar usuarios"

C.U. 51	<i>Editar usuarios</i>
Actores	Administrador
Descripción	Permite al administrador modificar los atributos de un usuario en la webapp.
Dependencias	<i>C.U.2, C.U.25</i>
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de usuarios 3. Selecciona el botón "Editar" para modificar el usuario deseado. 4. Modifica el nick,password,nombre,apellidos, email o el perfil del usuario. 5. Confirma la acción haciendo clic en el botón "Guardar".
Poscondición	Los cambios realizados en el nick,password,nombre,apellidos, email y perfi.se reflejan correctamente en la página de usuarios.
Excepciones	<ul style="list-style-type: none"> • En P4, se validan los datos del formulario, si alguno es incorrecto o falta algún dato obligatorio o si se ha duplicado el identificador del usuario se indicará con un mensaje de error dando al usuario la opción de subsanarlo.
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de gestión de usuarios

Tabla AI.52: Caso de uso 52 - “Eliminar usuarios”

C.U. 52	<i>Eliminar usuarios</i>
Actores	Administrador
Descripción	Permite al administrador eliminar un usuario en la aplicación..
Dependencias	C.U.2, C.U.48
Precondición	El usuario debe haber iniciado sesión y tener el rol de administrador.
Secuencia normal	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador en la webapp. 2. Accede al panel de usuarios 3. Selecciona el botón "Eliminar" en el usuario que deseas eliminar.Saldrá un pop up preguntando si estás seguro de eliminar esa usuario. 4. Confirma la acción haciendo clic en el botón "Si".
Poscondición	El usuario seleccionado se elimina de la base de datos, dejando de tener los privilegios de ser un usuario registrado, después vuelve a mostrar las opciones de búsqueda, alta, baja y modificación de usuario.
Excepciones	<ul style="list-style-type: none"> • En P3, si el administrador cancela la opción no se elimina ningún usuario y se vuelve a mostrar el panel de gestión de usuarios
Comentarios	- Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de gestión de usuarios

Anexo II

Detalle del diagrama de clases

La figura AII.1 muestra el diagrama de clases de la aplicación esquematizado, sin indicar las propiedades y métodos de cada clase (por razón de espacio), a continuación se detallan las propiedades y métodos de cada una de dichas clases para su mejor comprensión.

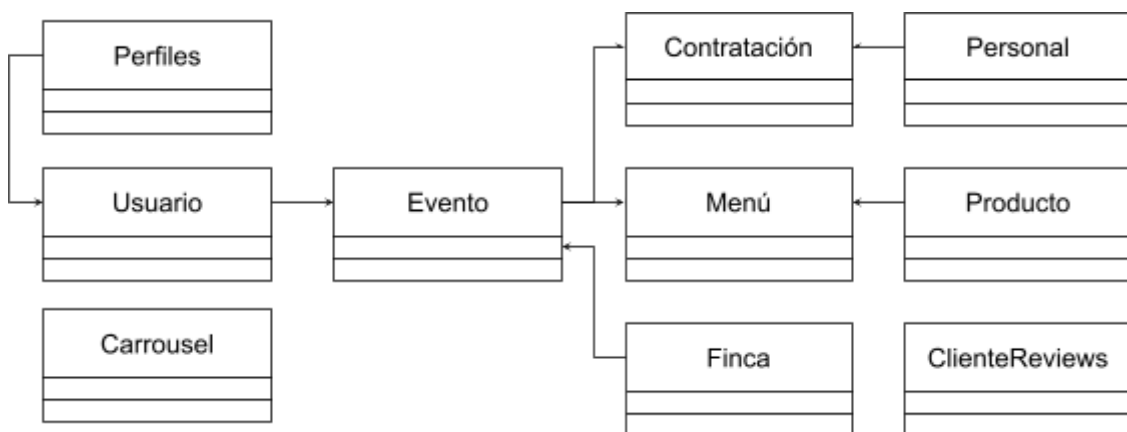


Figura AII.1: Diagrama de clases (esquema)

La clase “Usuario”:

- **Propiedades:**
 - id: int
 - nick: String
 - password: String
 - nombre: String
 - apellidos: String
 - activo: int
 - email: String
 - eventos: List<Evento>
- **Métodos:**
 - toString(): String
 - buscarTodos(): List<Usuario>
 - filtrarPorNombre(String nombre): List<Usuario>
 - filtrarPorNick(String username): Usuario
 - buscarPorId(Integer idCliente): Usuario
 - buscarPorNick(String username): int
 - existeNick(String nick): boolean
 - guardar(Usuario usuario): void
 - eliminar(int id): void

La clase “Perfiles”:

- **Propiedades:**
 - id: int
 - nick: String
 - perfil: String
- **Métodos:**
 - toString(): String
 - guardar(Perfiles perfil): void
 - buscarTodos(): List<Perfiles>
 - findPerfilByNick(String nick): String
 - buscarPorUsuario(Usuario usuario): Perfiles

La clase “Evento”:

- **Propiedades:**
 - idEvento: int
 - cliente: Usuario
 - idFinca: int

- fechaEvento: Date
- pedidoRealizado: int
- menús: List<Menu>
- contrataciones: List<Contratación>
- **Métodos:**
 - toString(): String
 - buscarTodas(): List<Evento>
 - buscarPorIdClienteEventoPendiente(int idCliente): List<Evento>
 - buscarPorFecha(Date fecha): List<Evento>
 - buscarEventoPorEstado(int estado): List<Evento>
 - buscarPorId(int idEvento): Evento
 - obtenerIdEventoPendientePorIdCliente(int idCliente): int
 - guardar(Evento evento): void
 - eliminar(int idEvento):void

La clase “Contratación”:

- **Propiedades:**
 - idContratación: int
 - idEvento: int
 - profesionPersonal: String
 - precioPersonal: int
 - evento: Evento
 - personal: Personal
- **Métodos:**
 - toString(): String
 - buscarTodas(): List<Contratación>
 - buscarPorId(int idPersonal): Contratación
 - obtenerContratacionesPorEvento(Evento evento): List<Contratacion>
 - guardar(Contratacion contrataciones): void
 - eliminar(int idPersonal): void

La clase “Personal”:

- **Propiedades:**
 - idPersonal: int
 - imagenPersonal: String
 - profesión: String
 - precio: int
 - precio_por_hora: int
 - horas: int
 - contrataciones: List<Contratación>

- Métodos:
 - toString():String
 - buscarTodas(): List<Personal>
 - buscarPorId(int idPersonal): Personal
 - guardar(Personal personal):void
 - eliminar(int idPersonal): void

La clase “Menu”:

- Propiedades:
 - idMenu: int
 - unidades: int
 - nombreProducto: String
 - precioProducto: Float
 - evento: Evento
 - producto: Producto
- Métodos:
 - toString(): String
 - buscarTodas(): List<Menu>
 - obtenerMenusPorEvento(Evento evento): List<Menu>
 - buscarPorId(Integer id): Menu
 - guardar(Menu menus): void
 - eliminar(int id): void

La clase “Producto”:

- Propiedades:
 - idProducto: Int
 - nombre: String
 - descripcion: String
 - imagenProducto: String
 - categoria: String
 - precio: Float
 - menús: List<Menu>
- Métodos:
 - toString(): String
 - buscarTodas(): List<Producto>
 - buscarPorCategoria(String categoria): List<Producto>
 - buscarPorId(Integer idProducto): Producto
 - guardar(Producto producto): void
 - eliminar(int idProducto): void

La clase “Finca”:

- **Propiedades:**
 - idFinca: int
 - imagenFinca: String
 - nombre: String
 - direccion: String
 - características: String
 - precio: int
- **Métodos:**
 - toString(): String
 - buscarTodas(): List<Finca>
 - buscarPorId(int idFinca): Finca
 - buscarFincasNoAsignadasEnFecha(Date fechaEvento): List<Finca>
 - guardar(Finca Finca): void
 - eliminar(int idFinca): void

La clase “Carrousel”:

- **Propiedades:**
 - id: int
 - carrouselImagen: String
 - titulo: String
 - subtítulo: String
- **Métodos:**
 - toString(): String
 - buscarTodas(): List<Carrousel>
 - buscarPorId(int idCarrousel): Carrousel
 - guardar(Carrousel carrousel): void
 - eliminar(int idCarrousel): void

La clase “ClienteReviews”:

- **Propiedades:**
 - idCliente: int
 - imagenCliente: String
 - nombreCliente: String
 - reseña: String
- **Métodos:**
 - toString(): String
 - buscarTodas(): List<ClienteReviews>
 - buscarPorId(int idCliente): ClienteReviews

- void guardar(ClienteReviews clienteReviews);
- void crear(ClienteReviews clienteReviews);

