



MEDAC

Instituto Oficial de Formación Profesional

[RentFit]

Desarrollo de Aplicaciones Web

Tutor: Javier Ruiz Jurado

Curso académico: 2025-2026

Autores

- Raul Henares
- Antonio Moyano
- Jose Ramón Rejano
- Javier Juárez

Noviembre, 2025

“Mens sana in corpore sano” [Décimo Junio Juvenal] (60-128)

Índice general

I Presentación	1
1. Introducción	3
1.1. Contexto del problema a resolver	3
1.2. Definición del problema real	3
1.3. Definición del problema técnico	4
1.3.1. Funcionamiento	4
1.3.2. Entorno	5
1.3.3. Vida esperada	5
1.3.4. Competencia	5
1.3.5. Aspecto externo	6
1.3.6. Estandarización	6
1.3.7. Calidad y fiabilidad	6
1.3.8. Programa de tareas	6
1.3.9. Pruebas	7
1.3.10 Seguridad	8
2. Objetivos	9
2.1. Objetivo principal	9
2.2. Objetivos específicos	9
2.3. Objetivos personales	11
3. Recursos	12
3.1. Recursos humanos	12
3.2. Recursos de hardware	12
3.3. Recursos de software	13

ÍNDICE GENERAL

II Contextualización empresarial	14
4. CONTEXTUALIZACIÓN EMPRESARIAL	16
4.1. Denominación	16
4.2. Justificación de la idea del proyecto	16
4.3. Estudio de la competencia	17
4.4. Oportunidad de negocio	17
4.5. Obligaciones fiscales	19
4.6. Financiación, ayudas y subvenciones	19
III Análisis	20
5. Especificación de requisitos	22
5.1. Introducción	22
5.2. Actores del sistema	22
5.3. Módulos de la aplicación	22
5.3.1. Módulo del usuario	23
5.3.2. Módulo del administrador	23
5.4. Requisitos del sistema	24
5.4.1. Requisitos funcionales	24
5.4.2. Requisitos no funcionales	25
5.4.3. Requisitos de la interfaz	26
5.4.4. Requisitos de información	26
6. Modelo de datos	28
6.1. Introducción	28
6.2. Tipos de entidad	28
6.2.1. Entidad Usuario	30
6.2.2. Entidad Rol	32
6.2.3. Entidad Dirección	34
6.2.4. Entidad Envio_Servicio	36
6.2.5. Entidad Contrato	38
6.2.6. Entidad Subscpcion	41

6.2.7. Entidad Tipo	42
6.2.8. Entidad Producto	44
6.2.9. Entidad Categorias	47
6.2.10Entidad Contrato_productos	49
6.3. Tipos de interrelación	50
6.3.1. Interrelación Usuario - Rol	52
6.3.2. Interrelación Usuario - Dirección	52
6.3.3. Interrelación Contrato - Subscpcion	53
6.3.4. Interrelación Subscpcion - Tipo	53
6.3.5. Interrelación Contrato - Contrato_productos	54
6.3.6. Interrelación Contrato_productos - Producto	54
6.3.7. Interrelación Categorias - Producto	54
6.4. Diagrama del Modelo Entidad-Interrelación	56
7. Análisis funcional	57
7.1. Introducción	57
7.2. Actores	57
7.3. Casos de uso	57
7.3.1. Caso de Uso 0. Diagrama de Contexto	58
7.3.2. CU-1.	59
7.4. Validación de casos de uso	60
7.5. Diagrama de secuencia	61
IV Diseño	62
8. Diseño de datos	64
8.1. Introducción	64
8.2. Modelo Relacional	64
8.2.1. Tabla Usuario	65
8.2.2. Tabla Categoria	65
8.2.3. Tabla Producto	66
8.2.4. Tabla Direccion	66

ÍNDICE GENERAL

8.2.5. Tabla Envio_servicio	66
8.2.6. Tabla Contrato	67
8.2.7. Tabla Contrato_productos	67
8.2.8. Tabla Tipo	67
8.2.9. Tabla Subscripcion	68
8.3. Normalización del modelo	68
8.3.1. Tabla Usuario	69
8.3.2. Tabla Rol	69
8.3.3. Tabla Categoria	69
8.3.4. Tabla Producto	70
8.3.5. Tabla Direccion	70
8.3.6. Tabla Envio_servicio	71
8.3.7. Tabla Contrato	71
8.3.8. Tabla Contrato_productos	72
8.3.9. Tabla Tipo	72
8.3.10. Tabla Subscripcion	73
8.4. Esquema relacional	74
8.5. Diagrama relacional	74
9. Diseño arquitectónico	76
9.1. Introducción	76
9.2. Diagrama de despliegue	76
9.2.1. Descripción de los nodos	76
9.2.2. Descripción de los componentes	76
10. Diseño de la interfaz	77
10.1. Introducción	77
10.2. Características comunes	77
10.3. Interfaz del módulo Usuario	77
10.3.1. Descripción general	77
10.3.2. Descripción detallada	77
10.3.3. Otros elementos de la interfaz	77
10.4. Interfaz del módulo Administrador	77

10.4.1Descripción general	77
10.4.2Descripción detallada	78
10.4.3Otros elementos de la interfaz	78
V Pruebas	79
11Pruebas	81
11.1Introducción	81
11.2Prueba de usuarios públicos	81
11.2.1Pruebas CU-1.1	81
VI Conclusiones y futuras mejoras	82
12Conclusiones	84
12.1Introducción	84
12.2Conclusiones específicas	84
12.3Conclusiones personales y profesionales	84
13Futuras mejoras	85
13.1Introducción	85
13.2Mejoras y nuevas funciones	85
13.2.1Mejoras para usuarios	85
13.2.2Mejoras para administrador	85
13.2.3Mejoras de funcionamiento	85
13.2.4Integración con otras aplicaciones	86
13.2.5Exportabilidad	86
14ANEXOS	87
BIBLIOGRAFÍA	87

Parte I

Presentación

Capítulo 1

Introducción

1.1. Contexto del problema a resolver

En la actualidad, la práctica deportiva y las actividades orientadas al bienestar físico han experimentado un crecimiento significativo. Este auge ha provocado que tanto gimnasios como centros deportivos, empresas dedicadas al acondicionamiento físico y usuarios particulares presenten una necesidad cada vez mayor de disponer de material de gimnasia actualizado, en buen estado y adaptado a sus necesidades.

Sin embargo, el acceso a este tipo de equipamiento supone un reto para muchas organizaciones debido a sus elevados costes, la rápida obsolescencia del material, los costes de mantenimiento asociados y la falta de flexibilidad en los modelos tradicionales de compra. Todo ello genera barreras económicas y logísticas que dificultan la renovación del material o incluso su adquisición inicial.

Este contexto ha impulsado la búsqueda de nuevos modelos que permitan una gestión más eficiente y accesible del material deportivo, promoviendo soluciones basadas en la suscripción y el alquiler, capaces de adaptarse a distintos perfiles de usuario.

1.2. Definición del problema real

Existe un problema generalizado con respecto al material de gimnasia, se dan muchos casos en que tanto empresas como gimnasios o particulares dependan de material obsoleto o ya muy desgastado o que incluso no puedan alcanzar a adquirir ciertos productos debido a que requieren una inversión demasiado elevada para su situación económica actual.

En resumen, el problema real que se desea resolver con el desarrollo de este proyecto es el siguiente: resolver el problema de que muchas empresas no se puedan permitir la adquisición de mucho material de gimnasia,

1.3. Definición del problema técnico

Para dar solución al problema identificado, se plantea el desarrollo de una aplicación web basada en un modelo de suscripción para el alquiler de material de gimnasia. Esta aplicación permitirá gestionar los servicios, controlar los diferentes niveles de suscripción y garantizar que los usuarios puedan acceder al material que necesiten.

Se utilizará una metodología de desarrollo incremental y modular, lo que permitirá construir la aplicación por fases, integrar las funcionalidades de manera progresiva y facilitar el mantenimiento y escalabilidad del sistema.

1.3.1. Funcionamiento

La aplicación web ofrecerá un sistema de gestión de servicios de suscripción mediante el cual los usuarios podrán alquilar material de gimnasia. El objetivo principal es proporcionar precios competitivos que permitan el acceso a cualquier tipo de cliente, desde empresas hasta usuarios particulares.

Los tipos de usuarios que interactuarán con la aplicación son los siguientes:

Los usuarios que interactuarán con la aplicación son los siguientes:

■ Administrador

- Estará registrado en el sistema y tendrá acceso a un panel de administración.
- Podrá gestionar el catálogo de material disponible, crear o modificar planes de suscripción, supervisar el estado del inventario y gestionar las cuentas de los usuarios registrados.
- Tendrá competencias exclusivas como la actualización de los datos del sistema, la revisión de reportes y el control de la facturación.

■ Usuario suscrito

- Podrá registrarse en la plataforma y seleccionar el tipo de suscripción más adecuado a sus necesidades.
- Tendrá acceso al material incluido en su plan y podrá solicitar cambios, ampliaciones o renovaciones.
- Podrá consultar su historial de alquileres, pagos y configuración de cuenta.

■ Usuario invitado

- Tendrá acceso limitado a la plataforma y podrá consultar la información general, los niveles de suscripción y el catálogo.
- No podrá alquilar material hasta completar el registro.

1.3.2. Entorno

Existirán dos entornos de trabajo para esta aplicación:

- Entorno de desarrollo y testeo: entorno que se utilizará para desarrollar, probar y depurar la aplicación web antes de su lanzamiento al entorno de producción.
- Entorno de producción: entorno en el que se encontrará la versión final de la aplicación web...

Existirán dos entornos de ejecución para esta aplicación.

- **Entorno de ejecución del administrador:** incluirá herramientas avanzadas de gestión, paneles de control y funciones exclusivas.
- **Entorno de ejecución del resto de usuarios:** diseñado para la interacción habitual con la aplicación, enfocado en la experiencia de alquiler y consulta de suscripciones.

La interfaz será un elemento clave del proyecto, actuando como medio de comunicación entre los usuarios y la aplicación web. Su diseño será intuitivo, accesible y responsive. Los requisitos específicos se detallarán en la Sección 5.4.3.

1.3.3. Vida esperada

El ciclo de vida esperado de la aplicación es elevado, ya que se pretende que el sistema pueda seguir ampliándose, integrándose con nuevas funcionalidades y adaptándose a las necesidades del mercado.

Teniendo en cuenta que el propósito de la creación de esta aplicación web es la realización de un proyecto, el mantenimiento de dicha aplicación no correrá a cargo de su autor.

No obstante, se realizará un diseño modular de cada una de las partes de la aplicación web para facilitar, en el futuro, posibles tareas de mantenimiento y mejoras.

1.3.4. Competencia

Actualmente existen pocos servicios relacionados con el alquiler de material deportivo, aunque la mayoría se centran en grandes empresas, equipamiento especializado o modelos más centrados en la compra que en el alquiler de material. Sin embargo, no son frecuentes las plataformas orientadas a gimnasios pequeños, empresas con recursos limitados o usuarios particulares que deseen acceder a un catálogo amplio mediante un sistema de suscripción flexible.

El proyecto pretende cubrir este nicho ofreciendo un servicio accesible, escalable y con precios adaptados a distintos niveles de uso.

1.3.5. Aspecto externo

En relación con el aspecto externo, se tendrán en cuenta los siguientes aspectos:

- **Interfaz de usuario**

- Se realizará una interfaz totalmente responsive, intuitiva y amigable para que el usuario pueda navegar de la manera más cómoda posible. El Capítulo 10 describirá el Diseño de la Interfaz de la aplicación web que se va a desarrollar.

- **Distribución de la aplicación: formato de almacenamiento**

- La aplicación podrá distribuirse a través de un servidor web alojado en un entorno local, facilitando su instalación y despliegue en distintos escenarios.

1.3.6. Estandarización

Para el desarrollo de la aplicación se seguirán las recomendaciones del *World Wide Web Consortium* (W3C) [3], que promueve estándares que favorecen la accesibilidad, la interoperabilidad y la calidad del software. Asimismo, se emplearán herramientas, frameworks y tecnologías que cuentan con el respaldo de organizaciones reconocidas y cumplen estándares internacionales.

También se debe tener en cuenta que los recursos que se van a utilizar son herramientas informáticas que están validadas por prestigiosas organizaciones que indican que cumplen con los estándares. Véase el capítulo 3 de Recursos.

1.3.7. Calidad y fiabilidad

La calidad y la fiabilidad de la aplicación web estarán garantizadas por:

- El uso de tecnologías estables, ampliamente documentadas y mantenidas por comunidades activas.
- La realización de pruebas exhaustivas unitarias, funcionales y de integración.
- Un diseño modular que facilite la detección y corrección de errores.
- La implementación de buenas prácticas de programación y arquitectura.

1.3.8. Programa de tareas

El desarrollo del presente proyecto va estar compuesto por la siguientes fases:

- Introducción: descripción del problema, establecimiento de los objetivos, revisión de antecedentes, identificación de restricciones iniciales y estratégicas y selección de recursos.
- Análisis: especificación de requisitos (funcionales, no funcionales, de información y de la interfaz), descripción del modelo de datos y análisis funcional (casos de uso y diagramas de secuencia).
- Diseño: descripción del diseño de datos, clases, paquetes y de la interfaz.
- Implementación: codificación de la aplicación web teniendo en cuenta el diseño desarrollado.
- Pruebas: comprobación de que la aplicación web funciona correctamente, es robusta y amigable.

1.3.9. Pruebas

La fase de pruebas es esencial para garantizar que la aplicación web funciona correctamente. En particular, se pretende comprobar que la aplicación web:

- Hace lo que debe hacer.
- No provoca efectos secundarios que pueden desencadenar situaciones catastróficas.
- Contiene módulos que se ejecutan correctamente.
- Garantiza los privilegios de cada tipo de usuario.

Cada prueba tendrá la siguiente estructura para detectar los errores y corregirlos:

- Objetivo de la prueba. Se debe indicar en qué consiste la prueba y el resultado esperado.
- Problema detectado, en su caso. Si ocurre un error entonces se debe describir la causa que lo ha provocado.
- Solución adoptada, en su caso. Si se ha producido un error, se deben indicar las medidas tomadas para solucionarlo.

El Capítulo 11 de Pruebas describirá las pruebas realizadas.

1.3.10. Seguridad

Hay varias consideraciones de seguridad que se deben tener en cuenta al desarrollar una aplicación web moderna. Algunas de las más importantes son:

- **Autenticación y autorización:** gestión segura de credenciales, control de accesos y privilegios.
- **Protección contra ataques de inyección:** filtrado de datos, consultas preparadas y validación estricta.
- **Cifrado de la información sensible:** uso de HTTPS y almacenamiento seguro de credenciales.
- **Protección frente a ataques de fuerza bruta:** limitación de intentos y uso de captchas.
- **Gestión segura de sesiones:** expiración automática, tokens seguros y protección contra secuestro de sesiones.

Capítulo 2

Objetivos

2.1. Objetivo principal

El objetivo principal de este proyecto es desarrollar una aplicación web que permita la gestión de servicios de suscripción para el alquiler de material de gimnasia. La plataforma ofrecerá un sistema flexible y accesible que facilite a gimnasios, otras empresas y usuarios particulares el acceso a equipamiento deportivo sin necesidad de realizar grandes inversiones iniciales. La aplicación deberá garantizar una experiencia de uso intuitiva y segura, permitiendo la administración del catálogo de productos, la gestión de usuarios y el control de los planes de suscripción.

2.2. Objetivos específicos

Los objetivos específicos de este proyecto son los siguientes:

■ Tipos de usuarios

Se deberán permitir los siguientes tipos de usuarios:

- **Administrador**

- Este usuario tendrá acceso completo al sistema y será responsable de la gestión del catálogo de material, los niveles de suscripción, los usuarios registrados y los aspectos administrativos de la plataforma.
- Podrá realizar tareas avanzadas como la actualización del inventario, la supervisión de pedidos, la resolución de incidencias y la administración de la información general del sistema.

- **Usuario registrado**

- Podrá acceder al sistema mediante registro y gestionar su suscripción.
- Tendrá acceso al catálogo de material disponible para su nivel de suscripción.

- Podrá consultar su historial, modificar su plan de suscripción o gestionar sus datos personales.

- **Usuario invitado**

- Podrá consultar información general, niveles de suscripción y catálogo básico.
- Tendrá funcionalidades limitadas hasta que complete el registro.

- **Base de datos relacional**

Se deberá diseñar una base de datos relacional que permita gestionar de forma eficiente toda la información relacionada con la aplicación. En particular:

- Gestión de usuarios registrados (administrador y usuarios suscritos).
- Gestión del catálogo de material deportivo disponible para alquiler.
- Gestión de niveles o planes de suscripción.
- Gestión de pedidos, renovaciones, devoluciones y estado del material.
- Registro de movimientos, operaciones y actividad dentro de la plataforma.

- **Módulos**

Se deberán diseñar los siguientes módulos principales:

- **Módulo del administrador**

- Gestión completa de usuarios.
- Gestión del catálogo de material deportivo.
- Administración de los niveles de suscripción.
- Revisión del estado del inventario y control del material disponible.
- Gestión de incidencias, solicitudes especiales y comunicación con usuarios.
- Consulta del registro de actividad del sistema.

- **Módulo del usuario registrado**

- Consulta del catálogo según su nivel de suscripción.
- Realización de solicitudes de alquiler y renovación.
- Gestión de su cuenta y suscripción.
- Acceso a soporte y documentación.

- **Módulo público**

- Consulta de información general sobre la plataforma.
- Consulta de los diferentes planes de suscripción disponibles.
- Acceso a preguntas frecuentes e información de ayuda.

- **Diseño de la interfaz**

Se deberá diseñar una interfaz intuitiva, sencilla y compatible con los distintos navegadores web. Además, será responsive para permitir su uso en móviles, tablets y ordenadores de escritorio. La interfaz deberá facilitar la navegación y ofrecer una buena experiencia para cualquier usuario.

2.3. Objetivos personales

Se proponen los siguientes objetivos personales:

- Aprender sobre el desarrollo de aplicaciones web.
- Profundizar en la construcción de bases de datos relacionales.
- Mejorar las habilidades de análisis, diseño y programación.
- Desarrollar buenas prácticas en seguridad, mantenimiento y escalabilidad de software.
- Lidiar con un proyecto complejo en su totalidad, desde el principio hasta el final.
- Tener a Raúl contento.

Capítulo 3

Recursos

3.1. Recursos humanos

Para el desarrollo del proyecto se ha contado con un equipo formado por cuatro integrantes, todos ayudando a los demás en sus apartados pero cada uno con un rol específico para asegurar una adecuada distribución del trabajo y una correcta ejecución de todas las fases del proyecto.

- **Raúl**

Responsable principal del backend y la lógica del servidor. Su labor incluye la creación de las API, la comunicación con la base de datos y la implementación de la seguridad del sistema.

- **Javier**

Responsable del análisis y la documentación técnica. Se encarga de la especificación de requisitos, la estructura de la documentación y la supervisión del seguimiento metodológico del proyecto.

- **Jose**

Responsable de frontend y diseño de la interfaz. Se encarga de la maquetación, la usabilidad, el diseño responsive y la experiencia de usuario.

- **Antonio**

Labores híbridas entre front y back apoyando a Raul y a Jose en gran medida con sus labores. Se encarga de unir backend y frontend, realizar pruebas, gestionar el despliegue en el servidor y solucionar incidencias del entorno.

3.2. Recursos de hardware

Para el desarrollo de la aplicación en el entorno local...:

- Ordenador: PC de sobremesa / portátil de uso personal.
- Sistema operativo: Windows 10 / Windows 11 / MacOs / Linux Ubuntu 22.04.
- RAM instalada: 16 GB.
- Procesador: Intel Core i5 / i7 de 10^a generación o superior, o equivalente AMD Ryzen.

Para el despliegue de la aplicación web, se hará uso de un servidor VPS con las características siguientes:

- Platform: Servidor VPS en la nube.
- OS Package: Linux Ubuntu Server 22.04 LTS.
- Memory: 2 GB – 4 GB de RAM.
- SSD: 40 GB – 80 GB de almacenamiento SSD.

3.3. Recursos de software

Se van a utilizar los siguientes recursos de software para el desarrollo de la aplicación web:

- Editor de código fuente: Visual Studio Code [2].
- Lenguaje backend: PHP 8.4 con framework Laravel.
- Lenguaje frontend: HTML5, CSS, JavaScript y React.
- Sistema gestor de bases de datos: MySQL.
- Control de versiones: Git y plataforma GitHub.
- Servidor local de desarrollo: XAMPP.
- Navegadores para pruebas: Google Chrome y Mozilla Firefox.
- Herramientas adicionales: Postman para pruebas de API, Figma para prototipado de interfaz, Composer para gestión de dependencias, plesk para el servidor.

Parte II

Contextualización empresarial

Capítulo 4

CONTEXTUALIZACIÓN EMPRESARIAL

4.1. Denominación

RentFit tiene la intención de cubrir las necesidades de particulares y empresas de disponer de material de gimnasia que puede llegar a resultar muy costoso a través de servicios de alquiler llevados a través de un sistema de suscripciones de distintos rangos

4.2. Justificación de la idea del proyecto

El origen de este proyecto surge de la evidente necesidad que presentan numerosos gimnasios, centros deportivos y entrenadores de acceder a material actualizado, de buena calidad y en óptimas condiciones. A esto se suma el elevado coste que suelen tener estos productos y la situación económica de muchos establecimientos, que dificulta su adquisición de manera permanente. Como resultado, estos negocios no siempre pueden asumir el impacto financiero que supone la compra constante de nuevo equipamiento. RentFit propone una alternativa innovadora frente a la compra definitiva de equipamiento deportivo, ofreciendo un servicio de alquiler flexible y accesible. Este modelo permite que gimnasios, centros deportivos y entrenadores puedan disponer de material actualizado y en buen estado sin tener que asumir el elevado coste de adquisición. De esta manera, RentFit facilita el acceso a recursos de calidad, reduce la carga económica de los establecimientos y contribuye a que puedan renovar o ampliar su equipamiento de forma más rápida, eficiente y sostenible. Además, el alquiler permite adaptar el material a las necesidades cambiantes de cada negocio, evitando inversiones innecesarias y mejorando su capacidad de respuesta ante variaciones en la demanda.

4.3. Estudio de la competencia

Procederéis a identificar quiénes son las empresas con proyectos en marcha que puedan considerarse competidores (hay que tener en cuenta el concepto de competencia en su sentido más amplio, es decir, no sólo existen competidores directos, sino que también podemos tener competidores indirectos). La competencia real a RentFit resulta componerse más por competencia indirecta que por competencia directa ya que existen pocas empresas que se dediquen al alquiler de este tipo de material que operen en Córdoba.

Competencia directa

Aunque limitada, sí existen algunas empresas que ofrecen servicios similares al modelo planteado por RentFit. Una de las más destacadas es FitnessRent, empresa dedicada al alquiler de equipamiento deportivo, principalmente para hoteles, anuncios publicitarios y eventos temporales.

Si bien su público objetivo difiere del de RentFit —centrado más en entornos corporativos, hoteleros o promocionales—, el hecho de ofrecer un servicio de alquiler de material deportivo la convierte en un competidor directo relevante. Representa un ejemplo claro de que existe demanda para un modelo basado en el acceso temporal al equipamiento sin necesidad de realizar una compra definitiva.

Competencia indirecta

La mayoría de competidores de RentFit pertenecen a la categoría de competencia indirecta. Se trata de empresas que no alquilan material, pero que ofrecen una alternativa válida mediante la venta directa de equipamiento deportivo, obligando al cliente a realizar una inversión inicial mayor.

Entre estas empresas destacan:

Decathlon

Sprinter

ProFit

Otros distribuidores especializados en material deportivo y fitness

Estas compañías disponen de una amplia variedad de productos y cuentan con presencia consolidada en el mercado. Sin embargo, su modelo de negocio exige a los gimnasios y entrenadores asumir el coste completo del equipamiento, lo que RentFit busca precisamente evitar ofreciendo un sistema más flexible y económicamente accesible.

4.4. Oportunidad de negocio

Oportunidad de negocio

El sector del fitness y la actividad física en España se encuentra en crecimiento

constante, tanto en número de usuarios como en demanda de servicios profesionales. RentFit surge como respuesta directa a una necesidad que va en aumento: la de disponer de material deportivo actualizado y en buen estado sin tener que asumir el elevado coste que implica la compra de equipamiento.

Demandas del mercado

Actualmente, muchos gimnasios, entrenadores personales y centros deportivos, especialmente los de menor tamaño, tienen limitaciones económicas que dificultan la renovación o ampliación de su equipamiento. Esta situación genera una demanda creciente de soluciones más flexibles y accesibles. El alquiler de material deportivo se presenta como una alternativa que:

Reduce la inversión inicial necesaria.

Permite adaptar el equipamiento a las necesidades del momento.

Evita la obsolescencia del material.

Facilita el acceso a productos de mayor calidad.

Aunque el modelo de alquiler en este sector aún no está ampliamente implantado en Córdoba, esta ausencia de competidores directos representa una oportunidad clara para introducir un servicio innovador con un mercado potencial significativo.

Viabilidad económica

Desde el punto de vista económico, RentFit resulta viable por varias razones:

Bajo nivel de competencia directa, lo que permite posicionarse como la alternativa principal en la ciudad.

Elevada demanda potencial, derivada del crecimiento del entrenamiento personal, centros boutique y gimnasios independientes.

Ingresos recurrentes mediante modelos de suscripción o alquiler mensual.

Rentabilidad del equipamiento, ya que el material puede generar beneficios de forma continua durante toda su vida útil.

Flexibilidad para escalar: es posible comenzar con un inventario reducido y ampliarlo según la demanda.

Conclusión

La combinación de una necesidad real, una competencia directa limitada y un modelo económico basado en ingresos recurrentes hace que RentFit represente una oportunidad de negocio sólida y con margen de crecimiento. Su propuesta responde a una carencia del mercado cordobés, ofreciendo una solución innovadora, rentable y adaptable a las necesidades cambiantes del sector fitness.

4.5. Obligaciones fiscales

Se debe especificar si el desarrollo del proyecto conlleva la necesidad de cumplimiento de obligaciones fiscales, laborales y/o de prevención de riesgos y sus condiciones de aplicación.

4.6. Financiación, ayudas y subvenciones

Habría que identificar las posibles necesidades de financiación para llevar a cabo el desarrollo del proyecto, así como la búsqueda de ayudas o subvenciones para la incorporación de las nuevas tecnologías de producción o de servicio propuestas.

Parte III

Análisis

Capítulo 5

Especificación de requisitos

5.1. Introducción

Se desea desarrollar una aplicación web que permita la gestión de servicios de suscripción para el alquiler de material de gimnasia. Esta plataforma estará orientada tanto a gimnasios como a empresas con recursos limitados y a usuarios particulares que deseen acceder a material deportivo sin realizar una inversión elevada.

Las siguientes secciones describen los actores del sistema (Sección 5.2), la descripción modular (Sección 5.3) y los requisitos del sistema (Sección 5.4).

5.2. Actores del sistema

Se consideran los siguientes tipos de usuario en la aplicación web:

- **Usuario invitado:** usuario no registrado que puede consultar información general sobre la plataforma, el catálogo básico y los niveles de suscripción.
- **Usuario registrado:** usuario que dispone de una cuenta activa y puede gestionar su suscripción, consultar material disponible según su plan, realizar solicitudes de alquiler o renovación y modificar su información personal.
- **Administrador:** usuario registrado con permisos avanzados con un control total sobre la aplicación. Será responsable de la gestión de usuarios, suscripciones, material, inventario y operaciones internas.

5.3. Módulos de la aplicación

La aplicación web estará compuesta por módulos correspondientes a cada tipo de usuario. En las siguientes subsecciones se detalla la funcionalidad de cada uno de ellos.

5.3.1. Módulo del usuario

El usuario registrado podrá realizar las siguientes acciones:

- Consultar el catálogo de material disponible según su nivel de suscripción.
- Solicitar el alquiler de material deportivo.
- Gestionar sus renovaciones o devoluciones.
- Modificar sus datos personales.
- Consultar su historial de uso y facturación.
- Acceder a soporte y documentación de ayuda.

El usuario invitado podrá:

- Consultar la información general de la plataforma.
- Visualizar los distintos niveles de suscripción.
- Consultar un catálogo básico del material disponible.
- Registrarse en la plataforma para acceder a las funcionalidades completas.

5.3.2. Módulo del administrador

El usuario administrador tendrá control total de la aplicación y será responsable de los siguientes módulos:

- Gestión del catálogo de material deportivo disponible.
- Gestión de usuarios registrados.
- Gestión de los niveles o planes de suscripción.
- Supervisión de pedidos, renovaciones y devoluciones.
- Control del inventario y del estado del material.
- Gestión de incidencias y solicitudes especiales.
- Acceso al registro de actividad de la aplicación.

En principio, la aplicación solo tendrá un único usuario con rol de administrador.

5.4. Requisitos del sistema

Los requisitos del sistema hacen referencia a todas las características relacionadas con la aplicación web. Se describirán los siguientes tipos de requisitos:

- Requisitos funcionales: describen las tareas que la aplicación debe realizar para satisfacer las necesidades del problema (Sección 5.4.1).
- Requisitos no funcionales: describen cómo se tiene que satisfacer las necesidades del problema (Sección 5.4.2).
- Requisitos de la interfaz: describen cómo debe ser la comunicación entre el usuario y la aplicación (Sección 5.4.3).
- Requisitos de la información: describen las características de los datos que se van a gestionar (Sección 5.4.4).

5.4.1. Requisitos funcionales

Los requisitos funcionales indican lo que el sistema debe hacer. Cada uno de estos requisitos debe tener dos propiedades:

- Ser completo: el requisito debe mencionar exactamente lo que el sistema debe hacer
- Ser cerrado: el requisito debe ser claro y no estar abierto a múltiples interpretaciones, sino solamente a una.

Los requisitos funcionales que se van a considerar se agruparán según los módulos de los tipos de usuario y se denominarán como RF-*n*, donde *n* es el número de requisito.

■ Módulo del usuario

La aplicación debe permitir que el usuario público pueda realizar las siguientes acciones:

- RF-1. Consultar el catálogo de material disponible.
- RF-2. Registrarse en la plataforma mediante un formulario.
- RF-3. Iniciar sesión con sus credenciales.
- RF-4. Seleccionar o modificar su plan de suscripción.
- RF-5. Solicitar el alquiler de material según su suscripción.
- RF-6. Consultar su historial de pedidos.
- RF-7. Gestionar renovaciones y devoluciones.
- RF-8. Modificar sus datos personales.

- RF-9. Consultar documentación de ayuda.

- **Módulo del administrador**

La aplicación debe permitir que el administrador pueda realizar las siguientes acciones:

- RF-10. Gestionar usuarios registrados.
 - RF-10.1. Crear un usuario.
 - RF-10.2. Buscar un usuario.
 - RF-10.3. Consultar un usuario.
 - RF-10.4. Modificar un usuario.
 - RF-10.5. Eliminar un usuario.
- RF-11. Gestionar el catálogo de material deportivo.
 - RF-11.1. Registrar nuevo material.
 - RF-11.2. Consultar material.
 - RF-11.3. Modificar información del material.
 - RF-11.4. Dar de baja material (borrado lógico).
- RF-12. Gestionar los planes de suscripción.
 - RF-12.1. Crear un plan de suscripción.
 - RF-12.2. Modificar un plan existente.
 - RF-12.3. Eliminar un plan (soft delete).
- RF-13. Supervisar pedidos de usuarios.
- RF-14. Gestionar incidencias reportadas por los usuarios.
- RF-15. Consultar reportes y estadísticas del sistema.

5.4.2. Requisitos no funcionales

Los requisitos no funcionales representan cómo tiene que trabajar la aplicación. Los requisitos no funcionales se denotarán como RNF-*n*, donde *n* es el número de requisito.

- RNF-1. La aplicación solo tendrá un usuario con rol de administrador.
- RNF-2. El borrado de registros se realizará mediante soft delete.
- RNF-3. La aplicación debe ser accesible desde cualquier navegador moderno.
- RNF-4. La interfaz debe ser completamente responsive.
- RNF-5. Las operaciones deben ejecutarse con un tiempo de respuesta inferior a 2 segundos.
- RNF-6. La aplicación debe seguir estándares de seguridad como HTTPS, hash de contraseñas y validación del lado del servidor.

- RNF-7. El sistema debe permitir una escalabilidad mínima para soportar al menos 500 usuarios simultáneos.
- RNF-8. La base de datos debe permitir integridad referencial y transacciones seguras.

5.4.3. Requisitos de la interfaz

La interfaz es el dispositivo que permite la comunicación entre el usuario y el sistema. En esta sección, se enumeran los requisitos que debe tener la interfaz para que pueda ser utilizada por todos los tipos de usuario.

Los requisitos de la interfaz especifican cómo debe ser la comunicación entre el usuario y la parte visible de la aplicación. Se denotarán como RINT-*n*, donde *n* es el número de requisito.

- RINT-1. La interfaz debe ser clara, intuitiva y coherente en todas las pantallas.
- RINT-2. Debe cumplir con los criterios básicos de accesibilidad WCAG.
- RINT-3. Debe permitir una navegación sencilla entre módulos.
- RINT-4. Los formularios deben incluir validaciones tanto visuales como textuales.
- RINT-5. Todos los mensajes de error deben mostrarse de forma visible y comprensible.
- RINT-6. El usuario debe recibir retroalimentación en cada acción (confirmaciones, avisos, alertas).

5.4.4. Requisitos de información

Los requisitos de información hacen referencia a los datos que debe gestionar la aplicación web. Se denotarán como RI-*n*, donde *n* es el número de requisito.

Se deberá almacenar la siguiente información:

- RI-1. Se debe almacenar la información personal de los usuarios (nombre, email, dirección, contraseña).
- RI-2. Se debe almacenar información de cada suscripción (nivel, fecha de inicio, renovación, estado).
- RI-3. Se debe registrar el catálogo completo de material deportivo disponible.
- RI-4. Se debe almacenar información sobre pedidos, devoluciones y renovaciones.
- RI-5. Se debe guardar el historial de actividad del administrador.

- RI-6. Se deben almacenar incidencias reportadas por los usuarios.

Una descripción más detallada de la información que se va a gestionar se puede consultar en el capítulo 6 de Modelo de Datos.

Capítulo 6

Modelo de datos

6.1. Introducción

En este capítulo se describirá conceptualmente el modelo de datos que empleará el sistema. Se identificarán las entidades que intervienen en el problema, sus atributos y las interrelaciones entre dichas entidades.

Para la confección del modelo, se empleará la notación del Modelo Entidad - Interrelación (modelo E-R) propuesto por Peter ChenEl esquema E-R describe la información representando los distintos elementos que la componen mediante un conjunto limitado de símbolos y reglas de relación entre ellos. Básicamente, los elementos principales del modelo son “tipo de entidad” y “tipo de interrelación”.

En las siguientes secciones se especifican con detalle los tipos de entidad y los tipos de interrelación que intervienen en el modelo, y se mostrará el diagrama E-R completo, que ofrecerá una visión global del problema.

6.2. Tipos de entidad

De acuerdo con el modelo E-R, un tipo de entidad representa a una serie de entes, objetos o personas reales o abstractos que forman parte del universo del problema a describir. Los tipos de entidad pueden ser fuertes o débiles.

- Tipo de entidad fuerte: su existencia no depende de la de otro tipo de entidad.
- Tipo de entidad débil: su existencia depende de la de otro tipo de entidad. La debilidad puede ser por existencia o por identificación.
 - Tipo de entidad débil por existencia: puede ser identificado por sí mismo a partir de sus atributos propios, pero requiere de la existencia de otro tipo de entidad del que depende.
 - Tipo de entidad débil por identificación: es un tipo de entidad débil por existencia que, además, requiere de algún atributo identificativo del tipo de entidad del que depende para poder ser identificado y diferenciado.

Las entidades de un determinado tipo de entidad se describen mediante un conjunto de atributos que representan cada una de las características o propiedades que lo describen.

Cada atributo tiene asociado un dominio de valores permitidos. Cada entidad se identifica y diferencia de forma inequívoca mediante un atributo identificador, que toma un valor único para cada entidad.

En esta sección se describirán todos los tipos de entidad que se han identificado que forman parte del problema descrito, indicando para cada uno de ellos la siguiente información:

- Descripción: definición de la entidad y función dentro del universo del problema.
- Restricción: indicación de si tiene alguna debilidad por identificación o existencia respecto de otra entidad.
- Características: se indicarán las siguientes:
 - Nombre del tipo de entidad.
 - Tipo: Fuerte o débil.
 - Atributos heredados.
 - Atributo identificador primario.
 - Atributo identificador alternativo.
 - Número de atributos, incluyendo los heredados.
- Atributos propios: por cada atributo, además del nombre del mismo, se indicará:
 - Definición: descripción del atributo.
 - Dominio: tipo de dato o valores que puede tomar el atributo.
 - Tipo: indicación de si es clave primaria o alterna, en su caso, atributo simple, etc.
 - Opcional: indicación de si el atributo puede contener un valor nulo o no.
 - Ejemplo: valor de muestra.
- Diagrama: representación gráfica del tipo de entidad, de acuerdo con la notación E-R.
- Ejemplo de entidad.

Los tipos de entidad que se han identificado y que se describirán a continuación son los siguientes:

- Tipo de entidad: Usuario
- Tipo de entidad: Rol

- Tipo de entidad: Dirección
- Tipo de entidad: Envio_Servicio
- Tipo de entidad: Contrato
- Tipo de entidad: Subscpcion
- Tipo de entidad: Tipo
- Tipo de entidad: Producto
- Tipo de entidad: Categorias
- Tipo de entidad: Contrato_products

6.2.1. Entidad Usuario

- **Descripción:** este tipo de entidad representa a los usuarios registrados en el sistema que pueden contratar servicios de suscripción, gestionar productos y realizar operaciones dentro de la plataforma. Cada usuario dispone de un rol que determina sus permisos de acceso y funcionalidades disponibles.
- **Restricciones:** es una entidad fuerte, por lo que no depende de ningún otro tipo de entidad para existir.
- **Características:**
 - Nombre del tipo de entidad: Usuario.
 - Tipo: Fuerte.
 - Atributos heredados: Ninguno.
 - Atributo identificador primario: id.
 - Atributo identificador alternativo: email.
 - Número de atributos: 8 propios.
- Atributos propios:
 - id
 - Definición: código numérico secuencial e incremental que identifica el usuario.
 - Dominio: números enteros mayores que 0.
 - Tipo: clave primaria.
 - Opcional: no
 - Ejemplo: 13
 - **nombre**
 - Definición: nombre del usuario.
 - Dominio: cadena de hasta 150 caracteres.
 - Tipo: atributo simple.

- Opcional: no.
- Ejemplo: Javier.

- **apellidos**

- Definición: apellidos del usuario.
- Dominio: cadena de hasta 150 caracteres.
- Tipo: atributo simple.
- Opcional: no.
- Ejemplo: Martínez López.

- **telefono**

- Definición: número de teléfono de contacto del usuario.
- Dominio: cadena numérica de entre 9 y 15 dígitos.
- Tipo: atributo simple.
- Opcional: no.
- Ejemplo: 678945321.

- **email**

- Definición: dirección de correo electrónico del usuario.
- Dominio: cadena de hasta 250 caracteres con formato email válido.
- Tipo: atributo simple / clave alternativa.
- Opcional: no.
- Ejemplo: usuario@example.com.

- **contraseña**

- Definición: contraseña encriptada utilizada para autenticar al usuario.
- Dominio: cadena de hasta 250 caracteres.
- Tipo: atributo simple.
- Opcional: no.
- Ejemplo: \$2y\$10\$XHfj23bd.... (hash).

- **fecha_alta**

- Definición: fecha en la que el usuario se registró en la plataforma.
- Dominio: tipo fecha.
- Tipo: atributo simple.
- Opcional: no.
- Ejemplo: 2025-02-10.

- **id_rol**

- Definición: identificador del rol que tiene asignado el usuario.
- Dominio: número entero positivo.
- Tipo: clave foránea.
- Opcional: no.
- Ejemplo: 1.

- **Diagrama (Figura 6.1):**

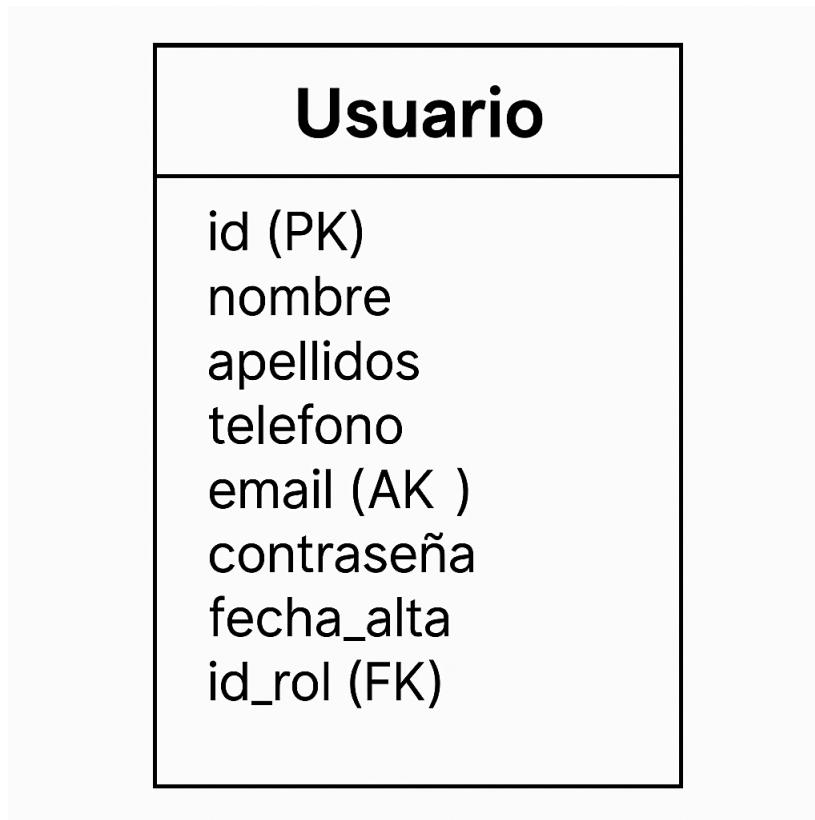


Figura 6.1: Entidad Usuario

- **Ejemplo de entidad (Tabla 6.1):**

Usuario	
Atributo	Valor
id	5
nombre	Antonio
apellidos	Núñez Ortega
telefono	665874321
email	antonio@ejemplo.com
contraseña	\$2y\$10\$7sd...
fecha_alta	2025-01-12
id_rol	2

Tabla 6.1: Ejemplo de la entidad *Usuario*

6.2.2. Entidad Rol

- **Descripción:** este tipo de entidad representa los distintos roles que pueden tener los usuarios dentro del sistema (por ejemplo, administrador o usuario estándar). El rol determina los permisos y funcionalidades a las que puede acceder cada usuario.

- **Restricciones:** es una entidad fuerte.

- **Características:**

- Nombre del tipo de entidad: Rol.
- Tipo: Fuerte.
- Atributos heredados: Ninguno.
- Atributo identificador primario: id.
- Atributo identificador alternativo: ninguno.
- Número de atributos: 2 propios.

- Atributos propios:

- **id**
 - Definición: identificador único del rol.
 - Dominio: números enteros mayores que 0.
 - Tipo: clave primaria.
 - Opcional: no.
 - Ejemplo: 1.
- **nombre**
 - Definición: nombre del rol asignado a los usuarios.
 - Dominio: cadena de hasta 100 caracteres.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: Administrador.

- **Diagrama (Figura 6.2):**

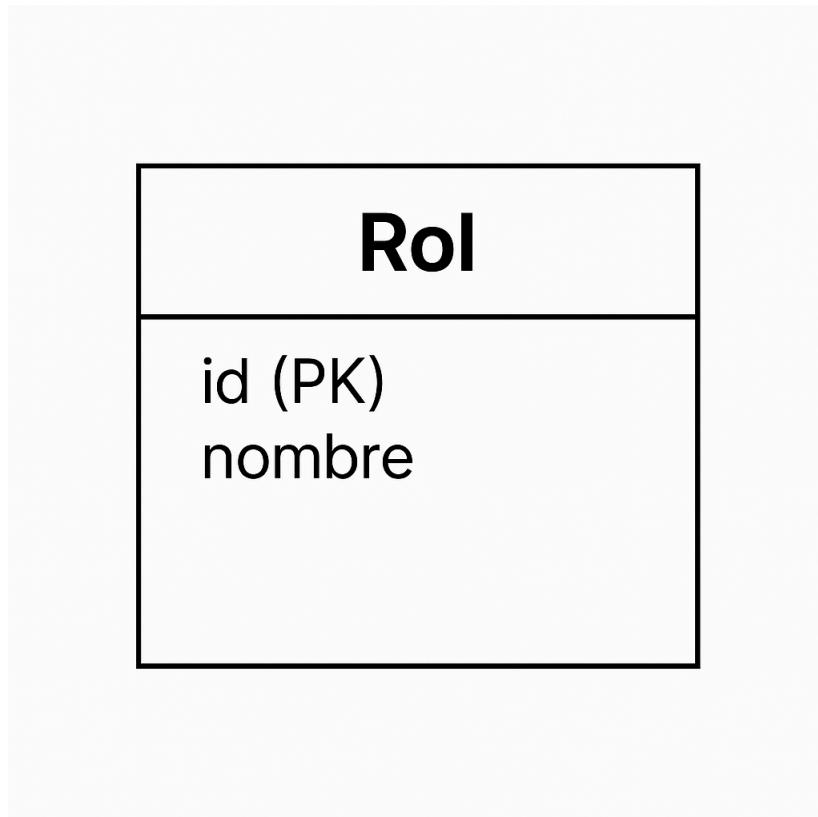


Figura 6.2: Entidad Rol

■ Ejemplo de entidad (Tabla 6.2):

Rol	
Atributo	Valor
id	1
nombre	Administrador

Tabla 6.2: Ejemplo de la entidad *Rol*

6.2.3. Entidad Dirección

- **Descripción:** este tipo de entidad almacena las direcciones asociadas a los usuarios, utilizadas para la entrega de material o prestación de servicios.
- **Restricciones:** es una entidad fuerte.
- **Características:**
 - Nombre del tipo de entidad: Dirección.
 - Tipo: Fuerte.
 - Atributos heredados: Ninguno.
 - Atributo identificador primario: id.
 - Atributo identificador alternativo: ninguno.

- Número de atributos: 5 propios.
- Atributos propios:
 - **id**
 - Definición: identificador único de la dirección.
 - Dominio: entero positivo.
 - Tipo: clave primaria.
 - Opcional: no.
 - Ejemplo: 3.
 - **id_usuario**
 - Definición: identificador del usuario al que pertenece la dirección.
 - Dominio: entero positivo.
 - Tipo: clave foránea.
 - Opcional: no.
 - Ejemplo: 5.
 - **vía**
 - Definición: nombre de la vía y número (calle, avenida, etc.).
 - Dominio: cadena de hasta 200 caracteres.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: Calle Mayor 15, 3ºB.
 - **cod_postal**
 - Definición: código postal de la dirección.
 - Dominio: cadena de 5 a 10 caracteres.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: 14004.
 - **Ciudad**
 - Definición: ciudad en la que se encuentra la dirección.
 - Dominio: cadena de hasta 100 caracteres.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: Córdoba.
- **Diagrama (Figura 6.3):**

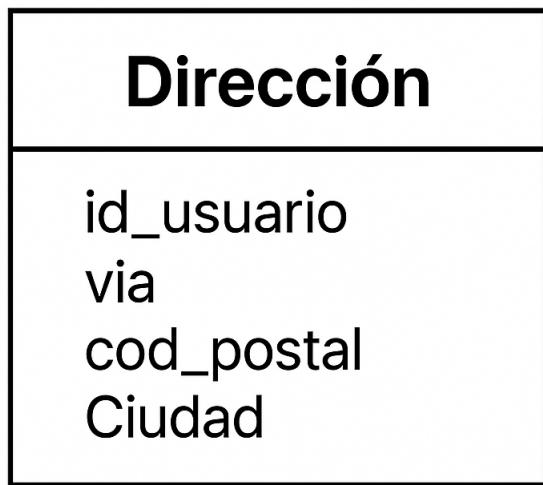


Figura 6.3: Entidad Dirección

- **Ejemplo de entidad (Tabla 6.3):**

Dirección	
Atributo	Valor
id	3
id_usuario	5
vía	Calle Mayor 15, 3ºB
cod_postal	14004
Ciudad	Córdoba

Tabla 6.3: Ejemplo de la entidad *Dirección*

6.2.4. Entidad Envio _ Servicio

- **Descripción:** este tipo de entidad representa los servicios de envío disponibles para una determinada dirección, incluyendo su coste y tiempo de envío.
- **Restricciones:** entidad fuerte.
- **Características:**
 - Nombre del tipo de entidad: Envio_Servicio.
 - Tipo: Fuerte.

- Atributos heredados: Ninguno.
 - Atributo identificador primario: id.
 - Atributo identificador alternativo: ninguno.
 - Número de atributos: 6 propios.
- Atributos propios:
- **id**
 - Definición: identificador único del servicio de envío.
 - Dominio: entero positivo.
 - Tipo: clave primaria.
 - Opcional: no.
 - Ejemplo: 2.
 - **nombre**
 - Definición: nombre comercial del servicio de envío.
 - Dominio: cadena de hasta 150 caracteres.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: Envío estándar 48h.
 - **tempo_envio**
 - Definición: tiempo estimado de envío.
 - Dominio: cadena de hasta 50 caracteres.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: 48 horas.
 - **coste_base**
 - Definición: coste base asociado al servicio de envío.
 - Dominio: número real positivo.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: 9.99.
 - **activo**
 - Definición: indica si el servicio de envío está disponible.
 - Dominio: valor booleano (true/false).
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: true.
 - **id_direccion**
 - Definición: identificador de la dirección asociada al servicio de envío.
 - Dominio: entero positivo.
 - Tipo: clave foránea.

- Opcional: no.
- Ejemplo: 3.

- **Diagrama (Figura 6.4):**

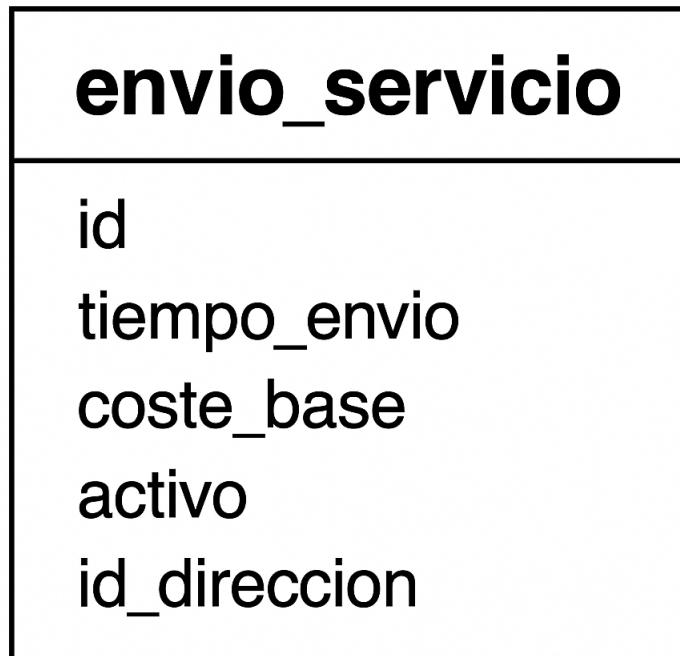


Figura 6.4: Entidad Envio_Servicio

- **Ejemplo de entidad (Tabla 6.4):**

Envio_Servicio	
Atributo	Valor
id	2
nombre	Envío estándar 48h
tempo_envio	48 horas
coste_base	9.99
activo	true
id_direccion	3

Tabla 6.4: Ejemplo de la entidad *Envio_Servicio*

6.2.5. Entidad Contrato

- **Descripción:** este tipo de entidad representa los contratos que realiza un usuario para la adquisición temporal de productos o servicios de suscripción.
- **Restricciones:** entidad fuerte.

■ Características:

- Nombre del tipo de entidad: Contrato.
- Tipo: Fuerte.
- Atributos heredados: Ninguno.
- Atributo identificador primario: id.
- Atributo identificador alternativo: ninguno.
- Número de atributos: 6 propios.

■ Atributos propios:**• *id***

- Definición: identificador único del contrato.
- Dominio: entero positivo.
- Tipo: clave primaria.
- Opcional: no.
- Ejemplo: 10.

• *id_usuario*

- Definición: identificador del usuario que realiza el contrato.
- Dominio: entero positivo.
- Tipo: clave foránea.
- Opcional: no.
- Ejemplo: 5.

• *id_direccion*

- Definición: identificador de la dirección asociada al contrato.
- Dominio: entero positivo.
- Tipo: clave foránea.
- Opcional: no.
- Ejemplo: 3.

• *fecha_inicio*

- Definición: fecha de inicio del contrato.
- Dominio: tipo fecha.
- Tipo: atributo simple.
- Opcional: no.
- Ejemplo: 2025-03-01.

• *fecha_fin*

- Definición: fecha de finalización del contrato.
- Dominio: tipo fecha.
- Tipo: atributo simple.
- Opcional: sí (en caso de contratos abiertos).
- Ejemplo: 2025-09-01.

- **descripcion**

- Definición: información adicional sobre el contrato.
- Dominio: texto de hasta 500 caracteres.
- Tipo: atributo simple.
- Opcional: sí.
- Ejemplo: Contrato de suscripción Premium para 6 meses.

- **Diagrama (Figura 6.5):**

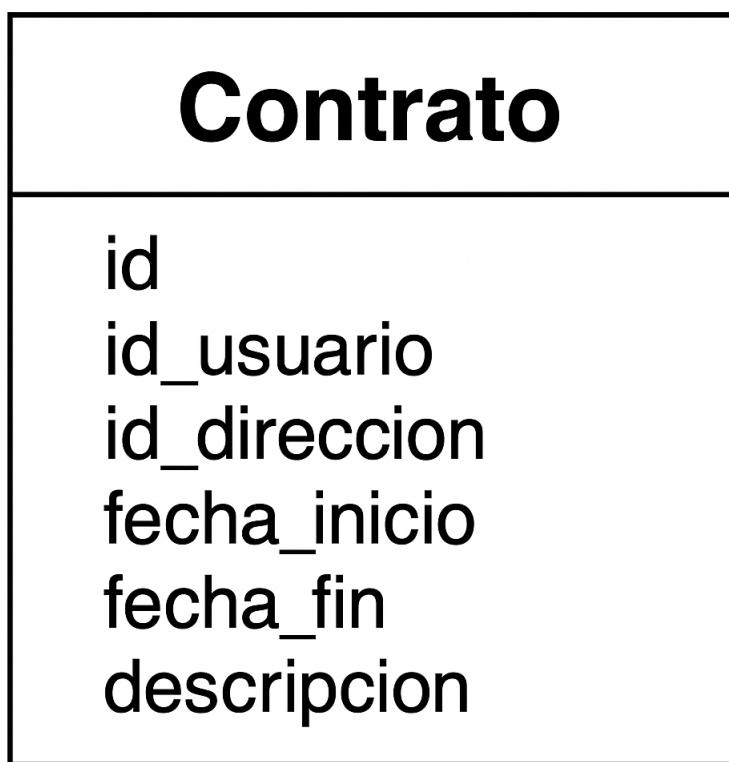


Figura 6.5: Entidad Contrato

- **Ejemplo de entidad (Tabla 6.5):**

Contrato	
Atributo	Valor
id	10
id_usuario	5
id_direccion	3
fecha_inicio	2025-03-01
fecha_fin	2025-09-01
descripcion	Contrato de suscripción Premium para 6 meses

Tabla 6.5: Ejemplo de la entidad *Contrato*

6.2.6. Entidad Subscpcion

- **Descripción:** este tipo de entidad representa las suscripciones activas o históricas asociadas a los contratos. Una suscripción define el tipo de plan contratado y su coste mensual.
- **Restricciones:** entidad fuerte.
- **Características:**
 - Nombre del tipo de entidad: Subscpcion.
 - Tipo: Fuerte.
 - Atributos heredados: Ninguno.
 - Atributo identificador primario: id.
 - Atributo identificador alternativo: ninguno.
 - Número de atributos: 4 propios.
- Atributos propios:
 - **id**
 - Definición: identificador único de la suscripción.
 - Dominio: entero positivo.
 - Tipo: clave primaria.
 - Opcional: no.
 - Ejemplo: 7.
 - **ID_tipo**
 - Definición: tipo de suscripción contratada.
 - Dominio: entero positivo.
 - Tipo: clave foránea.
 - Opcional: no.
 - Ejemplo: 2.
 - **ID_contrato**
 - Definición: identificador del contrato asociado.
 - Dominio: entero positivo.
 - Tipo: clave foránea.
 - Opcional: no.
 - Ejemplo: 10.
 - **mensualidad**
 - Definición: coste mensual de la suscripción.
 - Dominio: número real positivo.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: 29.99.

- **Diagrama (Figura 6.6):**

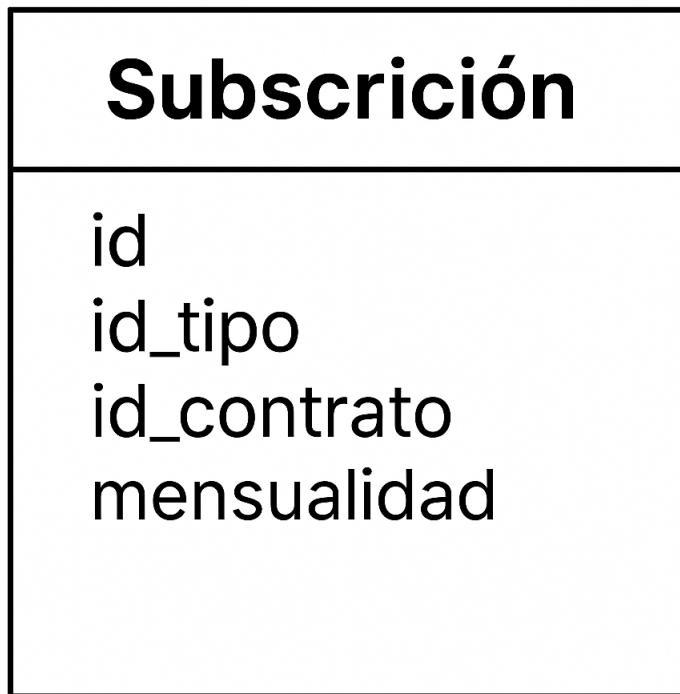


Figura 6.6: Entidad Subscricion

- **Ejemplo de entidad (Tabla 6.6):**

Subscricion	
Atributo	Valor
id	7
ID_tipo	2
ID_contrato	10
mensualidad	29.99

Tabla 6.6: Ejemplo de la entidad *Subscricion*

6.2.7. Entidad Tipo

- **Descripción:** este tipo de entidad define los tipos de suscripción disponibles (por ejemplo, Básica, Premium o Profesional), cada uno asociado a un precio, descripción y una imagen representativa.
- **Restricciones:** entidad fuerte.
- **Características:**
 - Nombre del tipo de entidad: Tipo.

- Tipo: Fuerte.
 - Atributos heredados: Ninguno.
 - Atributo identificador primario: ID.
 - Atributo identificador alternativo: ninguno.
 - Número de atributos: 4 propios.
- Atributos propios:
- **ID**
 - Definición: identificador único del tipo de suscripción.
 - Dominio: entero positivo.
 - Tipo: clave primaria.
 - Opcional: no.
 - Ejemplo: 2.
 - **Nombre**
 - Definición: nombre del tipo de suscripción.
 - Dominio: cadena de hasta 150 caracteres.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: Premium.
 - **Precio**
 - Definición: coste base mensual del tipo de suscripción.
 - Dominio: número real positivo.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: 29.99.
 - **Descripcion**
 - Definición: detalle del plan y sus beneficios.
 - Dominio: cadena de hasta 500 caracteres.
 - Tipo: atributo simple.
 - Opcional: sí.
 - Ejemplo: Acceso ilimitado a todo el catálogo.
 - **Imagen**
 - Definición: imagen asociada al tipo de suscripción.
 - Dominio: ruta o nombre de archivo.
 - Tipo: atributo simple.
 - Opcional: sí.
 - Ejemplo: premium.png.
- **Diagrama (Figura 6.7):**

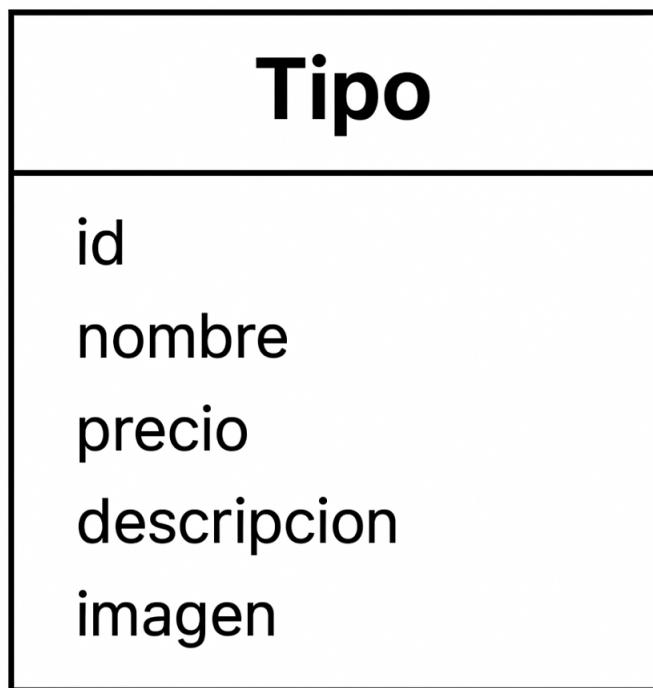


Figura 6.7: Entidad Tipo

- **Ejemplo de entidad (Tabla 6.7):**

Tipo	
Atributo	Valor
ID	2
Nombre	Premium
Precio	29.99
Descripcion	Acceso ilimitado a todo el catálogo
imagen	premium.png

Tabla 6.7: Ejemplo de la entidad *Tipo*

6.2.8. Entidad Producto

- **Descripción:** este tipo de entidad representa los productos deportivos disponibles para alquiler, con información como nombre, categoría, cantidad, imagen y características adicionales.
- **Restricciones:** entidad fuerte.
- **Características:**
 - Nombre del tipo de entidad: Producto.

- Tipo: Fuerte.
 - Atributos heredados: Ninguno.
 - Atributo identificador primario: ID.
 - Atributo identificador alternativo: ninguno.
 - Número de atributos: 8 propios.
- Atributos propios:
- **ID**
 - Definición: identificador único del producto.
 - Dominio: entero positivo.
 - Tipo: clave primaria.
 - Opcional: no.
 - Ejemplo: 15.
 - **ID_Categoría**
 - Definición: categoría a la que pertenece el producto.
 - Dominio: entero positivo.
 - Tipo: clave foránea.
 - Opcional: no.
 - Ejemplo: 3.
 - **Nombre**
 - Definición: nombre del producto.
 - Dominio: cadena de hasta 150 caracteres.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: Mancuernas 10 kg.
 - **Descripcion**
 - Definición: información descriptiva del producto.
 - Dominio: hasta 500 caracteres.
 - Tipo: atributo simple.
 - Opcional: sí.
 - Ejemplo: Mancuernas recubiertas de goma antideslizante.
 - **Cantidad**
 - Definición: número de unidades disponibles.
 - Dominio: entero positivo.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: 20.
 - **Instalacion**
 - Definición: indica si el producto requiere instalación.
 - Dominio: booleano.

- Tipo: atributo simple.
- Opcional: no.
- Ejemplo: false.
- **espacio**
 - Definición: espacio aproximado necesario para el producto.
 - Dominio: cadena de hasta 50 caracteres.
 - Tipo: atributo simple.
 - Opcional: sí.
 - Ejemplo: 1 m².
- **Precio**
 - Definición: precio del alquiler mensual.
 - Dominio: número real positivo.
 - Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: 12.99.
- **Imagen**
 - Definición: imagen ilustrativa del producto.
 - Dominio: cadena con ruta o nombre de archivo.
 - Tipo: atributo simple.
 - Opcional: sí.

■ **Diagrama (Figura 6.8):**

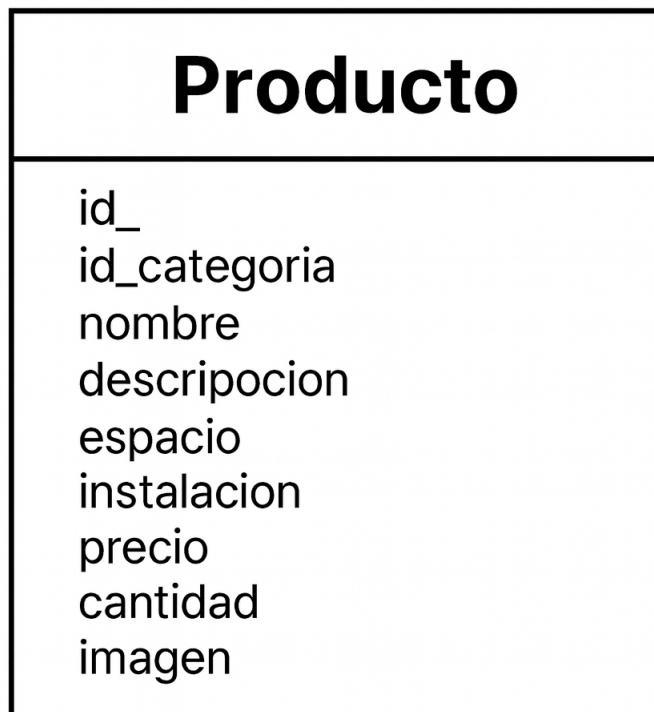


Figura 6.8: Entidad Producto

■ **Ejemplo de entidad (Tabla 6.8):**

Producto	
Atributo	Valor
ID	15
ID_Categoría	3
Nombre	Mancuernas 10 kg
Descripcion	Mancuernas recubiertas de goma antideslizante
Cantidad	20
Instalacion	false
espacio	1 m ²
Precio	12.99
Imagen	imagen

Tabla 6.8: Ejemplo de la entidad *Producto*

6.2.9. Entidad Categorias

- **Descripción:** este tipo de entidad representa las categorías del catálogo de productos. Las categorías pueden estar organizadas jerárquicamente mediante el atributo `padre_id`.
- **Restricciones:** entidad fuerte.
- **Características:**
 - Nombre del tipo de entidad: Categorias.
 - Tipo: Fuerte.
 - Atributos heredados: Ninguno.
 - Atributo identificador primario: ID.
 - Atributo identificador alternativo: ninguno.
 - Número de atributos: 3 propios.
- Atributos propios:
 - **ID**
 - Definición: identificador único de la categoría.
 - Dominio: entero positivo.
 - Tipo: clave primaria.
 - Opcional: no.
 - Ejemplo: 3.
 - **Nombre**
 - Definición: nombre de la categoría.
 - Dominio: cadena de 150 caracteres.

- Tipo: atributo simple.
 - Opcional: no.
 - Ejemplo: Pesas libres.
- **padre_id**
 - Definición: referencia a la categoría padre (si existe jerarquía).
 - Dominio: entero positivo o nulo.
 - Tipo: clave foránea opcional.
 - Opcional: sí.
 - Ejemplo: null.

- **Diagrama (Figura 6.9):**

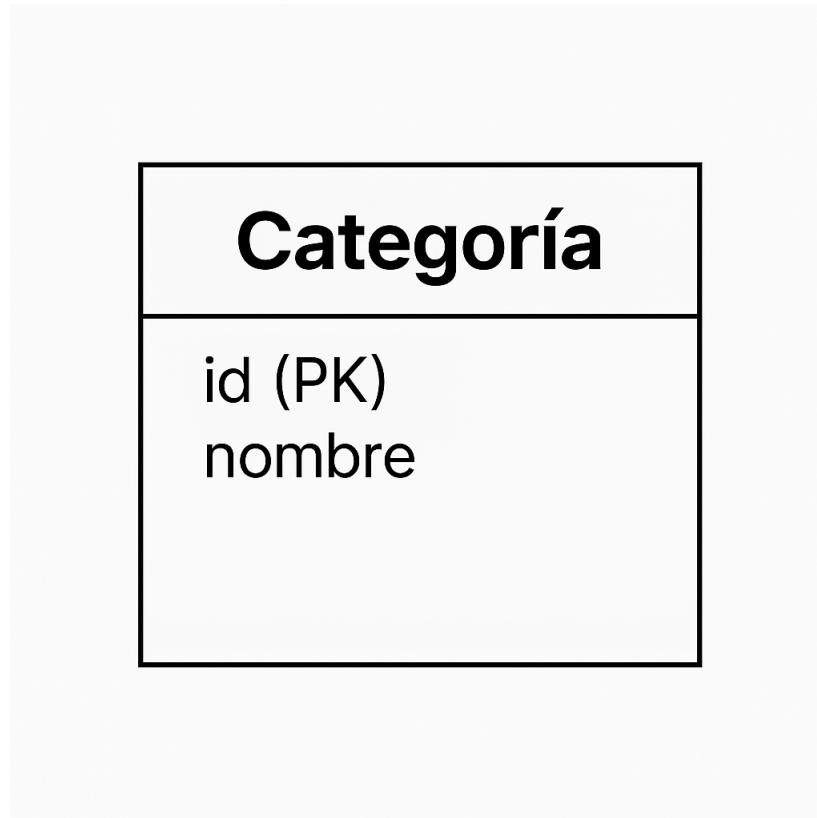


Figura 6.9: Entidad Categorias

- **Ejemplo de entidad (Tabla 6.9):**

Categorias	
Atributo	Valor
ID	3
Nombre	Pesas libres
padre_id	null

Tabla 6.9: Ejemplo de la entidad *Categorias*

6.2.10. Entidad Contrato_products

- **Descripción:** esta entidad representa la relación entre un contrato y los productos asociados al mismo. Permite registrar qué productos forman parte de cada contrato y su coste mensual individual.
- **Restricciones:** entidad fuerte (aunque conceptualmente actúa como una entidad intermedia de relación N:N).
- **Características:**
 - Nombre del tipo de entidad: Contrato_products.
 - Tipo: Fuerte.
 - Atributos heredados: Ninguno.
 - Atributo identificador primario: ID.
 - Atributo identificador alternativo: ninguno.
 - Número de atributos: 4 propios.
- Atributos propios:
 - **ID**
 - Definición: identificador único del registro de producto asociado a contrato.
 - Dominio: entero positivo.
 - Tipo: clave primaria.
 - Opcional: no.
 - Ejemplo: 40.
 - **ID_producto**
 - Definición: identificador del producto asociado al contrato.
 - Dominio: entero positivo.
 - Tipo: clave foránea.
 - Opcional: no.
 - Ejemplo: 15.
 - **ID_contrato**
 - Definición: identificador del contrato.
 - Dominio: entero positivo.
 - Tipo: clave foránea.
 - Opcional: no.
 - Ejemplo: 10.
 - **mensualidad**
 - Definición: coste mensual del producto dentro del contrato.
 - Dominio: número real positivo.
 - Tipo: atributo simple.
 - Opcional: no.

- Ejemplo: 12.99.

- **Diagrama (Figura 6.10):**



Figura 6.10: Entidad Contrato_productos

- **Ejemplo de entidad (Tabla 6.10):**

Contrato_productos	
Atributo	Valor
ID	40
ID_producto	15
ID_contrato	10
mensualidad	12.99

Tabla 6.10: Ejemplo de la entidad *Contrato_productos*

6.3. Tipos de interrelación

En esta sección se identificarán y describirán las interrelaciones entre los tipos de entidad descritos en la sección 7.2.

Las interrelaciones pueden ser de tipo débil o fuerte.

- Un tipo de Interrelación Fuerte es aquella que representa la relación existente entre dos tipos de entidad fuertes.

- Un tipo de Interrelación Débil es aquella que representa la relación entre un tipo de entidad débil y otro fuerte o entre dos tipos de entidad débiles.

De acuerdo con la notación del modelo E-R, una interrelación se representa mediante un rombo del que parten flechas hacia los tipos de entidad que forman parte de la relación. Cada tipo de entidad interviene en la interrelación con una determinada cardinalidad, que indica el número mínimo y máximo de instancias de cada tipo de entidad que pueden participar en la interrelación. Se representa por dos valores entre paréntesis (mínimo y máximo). Las posibles cardinalidades son: (0,1), (1,1),(0,n),(1,n),(m,n). Estas cardinalidades determinan el tipo de interrelación que se está definiendo, que puede ser:

- **1:1** Uno a uno. Cuando los dos tipos de entidad participan con una cardinalidad máxima de 1.
- **1:N o N:1** Uno a muchos, o muchos a uno. Cuando uno de los tipos de entidad participa con una cardinalidad máxima de 1 y el otro con una cardinalidad máxima de n.
- **N:N** Muchos a muchos. Cuando ambos tipos de entidad participan una cardinalidad máxima de n.

Para cada una de las interrelaciones que se han identificado en la definición del problema, se indicará la siguiente información:

- **Nombre.** Nombre del tipo de interrelación.
- **Descripción.** Definición del tipo de interrelación y de los tipos de entidad que participan en ella.
- **Tipo.** Indicación de si se trata de un tipo de interrelación débil o fuerte identificando los tipos de debilidad en su caso.
- **Cardinalidad.** Indicación de la cardinalidad del tipo de interrelación y cardinalidades mínima y máxima de los tipos de entidad intervenientes.
- **Atributos.** Indicación del número y descripción de los atributos del tipo de interrelación, en su caso.
- **Diagrama.** Representación gráfica del tipo de interrelación, de acuerdo con la notación E-R.
- **Ejemplo.** Valores de muestra.

Se han identificado las interrelaciones que se indican a continuación y que se describirán en las siguientes subsecciones:

- Tipo de interrelación: Usuario - Rol

- Tipo de interrelación: Usuario - Dirección
- Tipo de interrelación: Dirección - Envio_Servicio
- Tipo de interrelación: Usuario - Contrato
- Tipo de interrelación: Dirección - Contrato
- Tipo de interrelación: Contrato - Subscpcion
- Tipo de interrelación: Subscpcion - Tipo
- Tipo de interrelación: Contrato - Contrato_productos
- Tipo de interrelación: Contrato_productos - Producto
- Tipo de interrelación: Categorias - Producto

6.3.1. Interrelación Usuario - Rol

- **Nombre.** Usuario - Rol.
- **Descripción.** Relación que vincula a cada usuario con el rol que tiene asignado en el sistema. Un usuario desempeña exactamente un rol, mientras que un rol puede estar asociado a varios usuarios.
- **Tipo.** Interrelación fuerte entre dos entidades fuertes (Usuario y Rol).
- **Cardinalidad.** Un rol puede estar asociado a entre 0 y N usuarios ($0, N$). Cada usuario pertenece exactamente a un único rol (1, 1). Por tanto, se trata de una interrelación de tipo $N : 1$ (muchos usuarios a un rol).
- **Atributos.** La interrelación no tiene atributos propios; la relación se identifica mediante las claves ajenas en la entidad Usuario (id_rol).
- **Diagrama.** Se representa mediante un rombo que conecta las entidades Usuario y Rol.
- **Ejemplo.** Un usuario con id = 5, nombre = “Antonio”, pertenece al rol con id = 1, nombre = “Administrador”.

6.3.2. Interrelación Usuario - Dirección

- **Nombre.** Usuario - Dirección.
- **Descripción.** Relación que indica qué direcciones pertenecen a cada usuario. Un usuario puede registrar varias direcciones, mientras que cada dirección está asociada a un único usuario.
- **Tipo.** Interrelación fuerte entre las entidades Usuario y Dirección.

- **Cardinalidad.** Un usuario puede tener entre 0 y N direcciones ($0, N$). Cada dirección pertenece exactamente a un usuario (1, 1). Se trata de una relación de tipo $1 : N$ (un usuario, muchas direcciones).
- **Atributos.** La interrelación no tiene atributos propios; se materializa mediante el atributo `id_usuario` en la entidad Dirección.
- **Diagrama.** Representada mediante un rombo que conecta Usuario y Dirección.
- **Ejemplo.** El usuario con `id = 5` tiene asociadas las direcciones con `id = 3` (Calle Mayor 15) y `id = 4` (Avenida del Deporte 2).

6.3.3. Interrelación Contrato - Subscpcion

- **Nombre.** Contrato - Subscpcion.
- **Descripción.** Relación que vincula los contratos con las suscripciones que incluyen. Un contrato puede estar compuesto por una o varias suscripciones.
- **Tipo.** Interrelación fuerte entre Contrato y Subscpcion.
- **Cardinalidad.** Un contrato puede tener entre 0 y N suscripciones ($0, N$). Cada suscripción pertenece exactamente a un contrato (1, 1). Es una interrelación $1 : N$.
- **Atributos.** Sin atributos propios; se refleja mediante `ID_contrato` en Subscpcion.
- **Diagrama.** Rombo entre Contrato y Subscpcion.
- **Ejemplo.** El contrato con `id = 10` incluye la suscripción con `id = 7`.

6.3.4. Interrelación Subscpcion - Tipo

- **Nombre.** Subscpcion - Tipo.
- **Descripción.** Relación que indica de qué tipo es cada suscripción (por ejemplo, Básica, Premium, etc.).
- **Tipo.** Interrelación fuerte entre Subscpcion y Tipo.
- **Cardinalidad.** Un tipo de suscripción puede estar asociado a entre 0 y N suscripciones ($0, N$). Cada suscripción pertenece exactamente a un tipo (1, 1). Es una relación $N : 1$.
- **Atributos.** Sin atributos propios; se implementa mediante `ID_tipo` en Subscpcion.
- **Diagrama.** Rombo entre Subscpcion y Tipo.
- **Ejemplo.** La suscripción con `id = 7` pertenece al tipo con `ID = 2` (Premium).

6.3.5. Interrelación Contrato - Contrato_products

- **Nombre.** Contrato - Contrato_products.
- **Descripción.** Relación que asocia un contrato con los distintos productos incluidos en él. Cada contrato puede tener varios registros en Contrato_products.
- **Tipo.** Interrelación fuerte entre Contrato y Contrato_products.
- **Cardinalidad.** Un contrato puede tener entre 0 y N filas en Contrato_products (0, N). Cada registro de Contrato_products pertenece a un único contrato (1, 1). Es una interrelación 1 : N.
- **Atributos.** Los atributos propios de la relación (como mensualidad) se almacenan en la entidad Contrato_products.
- **Diagrama.** Rombo entre Contrato y Contrato_products.
- **Ejemplo.** El contrato con id = 10 tiene los registros Contrato_products con ID = 40 y 41.

6.3.6. Interrelación Contrato_products - Producto

- **Nombre.** Contrato_products - Producto.
- **Descripción.** Relación que indica qué producto concreto está asociado a cada registro de Contrato_products.
- **Tipo.** Interrelación fuerte entre Contrato_products y Producto.
- **Cardinalidad.** Un producto puede formar parte de entre 0 y N registros en Contrato_products (0, N). Cada registro de Contrato_products hace referencia a un único producto (1, 1). Es una interrelación N : 1.
- **Atributos.** Sin atributos adicionales; la mensualidad está en Contrato_products.
- **Diagrama.** Rombo entre Contrato_products y Producto.
- **Ejemplo.** El registro Contrato_products con ID = 40 hace referencia al producto con ID = 15.

6.3.7. Interrelación Categorias - Producto

- **Nombre.** Categorias - Producto.
- **Descripción.** Relación que establece la categoría a la que pertenece cada producto del catálogo.
- **Tipo.** Interrelación fuerte entre Categorias y Producto.

- **Cardinalidad.** Una categoría puede tener entre 0 y N productos ($0, N$). Cada producto pertenece exactamente a una categoría (1, 1). Es una interrelación $1 : N$.
- **Atributos.** No tiene atributos propios; se modela mediante ID_Categoría en Producto.
- **Diagrama.** Rombo entre Categorías y Producto.
- **Ejemplo.** La categoría con ID = 3 (Pesas libres) contiene el producto con ID = 15 (Mancuernas 10 kg).

6.4. Diagrama del Modelo Entidad-Interrelación

El diagrama completo se muestra en la figura 6.11

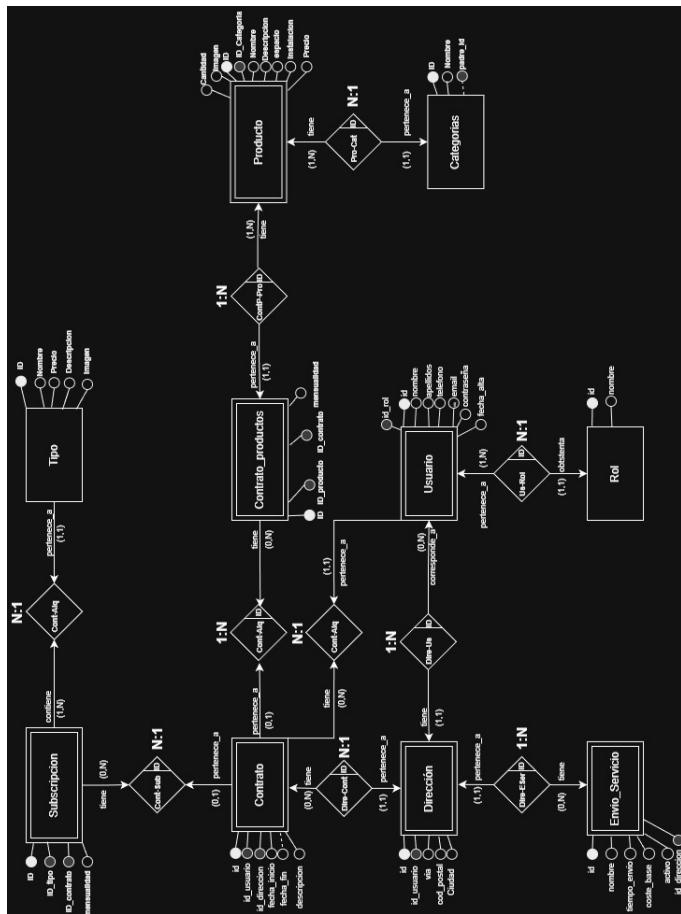


Figura 6.11: Diagrama del modelo Entidad - Interrelación

Capítulo 7

Análisis funcional

7.1. Introducción

Este capítulo identificará a los tipos de actores que podrán interactuar con el sistema informático que se desea desarrollar. A continuación, se utilizarán los casos de uso y los diagramas de secuencia para describir las acciones que podrán realizar los diferentes actores.

7.2. Actores

Un actor es una representación de una persona, proceso o entidad externa que interactúa con el sistema. Se van a considerar los siguientes tipos de actores:

- Administrador
 - Este tipo de usuario estará registrado en el sistema y tendrá un control completo de la aplicación.
 - En particular, tendrá las competencias exclusivas de la gestión de...

7.3. Casos de uso

Los casos de uso describen las acciones que pueden desarrollar los actores del sistema. Se ha identificado los siguientes casos de uso principales, que son descritos en las secciones que se indican:

- **CU-0. Diagrama de contexto** (Sección 7.3.1).
- **CU-1. Administrar...** (Sección 7.3.2).

7.3.1. Caso de Uso 0. Diagrama de Contexto

El diagrama de contexto engloba los casos de uso principales que componen el sistema. Véanse la Figura 7.1 y la Tabla 7.1.

Figura 7.1: Diagrama de Contexto

Tabla 7.1: CU-0. Diagrama de contexto

Caso de uso 0 - Diagrama de contexto	
Actores	Administrador
Descripción	Acciones principales del sistema
Precondiciones	El usuario debe acceder a la página inicial de la aplicación
Casos de uso	CU-1. CU-2. Administrar
Flujo principal	1a. El usuario público accede a la aplicación sin necesidad de iniciar sesión y podrá. 1b. El resto de usuarios se deberá identificar para acceder a los módulos correspondientes dentro de la aplicación.
Flujo alternativo	1. La aplicación no se carga correctamente. 2. Se informa del error al usuario. 3. Se informa al usuario que puede intentar cargar de nuevo la aplicación.

7.3.2. CU-1.

El caso de uso *CU-1.* describe las acciones que puede realizar (véanse la Tabla 7.2 y la Figura 7.2.). Este caso de uso está compuesto por los siguientes sub-casos de uso que se describen en las tablas que se indican:

- CU-1.1

Figura 7.2: CU-1.

Tabla 7.2: CU-1.

Caso de uso 1 -	
Actores	Usua
Descripción	Acciones que puede realizar el usuario
Precondiciones	El usuario debe haber accedido a la página principal de la aplicación y elegido el caso de uso CU-1..
Casos de uso	CU-1.1. CU-1.2. Administrar
Flujo principal	1. El usuario elige un 2. El sistema muestra
Flujo alternativo	1. Se produce un error al intentar ejecutar la información elegida. 2. Se informa al usuario del error. 3. El sistema vuelve a mostrar las opciones disponibles.

7.4. Validación de casos de uso

La tabla 7.3 permite comprobar que los casos de uso cubren todos los requisitos funcionales de la aplicación web propuestos en la Sección 5.4.1.

Tabla 7.3: Tabla validación casos de uso

Requisito funcional	Caso de uso
RF-1	CU 1.1
RF-8	CU 2.1

7.5. Diagrama de secuencia

El diagrama de secuencia es una representación gráfica que pretende dar una visión de las acciones que se realizarán durante la ejecución de alguna operación en el sistema. A continuación, se muestran los diagramas de secuencias para la creación (Figura 7.3), búsqueda (Figura 7.5), modificación (Figura 7.6) y borrado (Figura 7.4) de instancias genéricas () que se corresponderán con...

Figura 7.3: Diagrama de secuencia de crear

Figura 7.4: Diagrama de secuencia de borrar

Figura 7.5: Diagrama de secuencia de buscar

Figura 7.6: Diagrama de secuencia de modificar

Parte IV

Diseño

Capítulo 8

Diseño de datos

8.1. Introducción

En este capítulo se definirán las estructuras de datos que conforman el sistema a partir de los elementos identificados durante el análisis de datos del Capítulo 7 (tipos de entidad y tipos de interrelación). Para ello, se llevarán a cabo los siguientes pasos:

- Obtención del modelo relacional. Definición de la estructuras (tablas) del modelo de datos.
- Normalización del modelo. Refinamiento del modelo, para la eliminación de errores de integridad.
- Obtención del esquema relacional.
- Diagrama relacional.
-

8.2. Modelo Relacional

A partir del Modelo Entidad - Interrelación descrito en el capítulo 7, se pueden obtener las tablas o relaciones del Modelo Relacional utilizando las reglas de transformación (RTECAR - véase el capítulo 5 de Bases de Datos. Desde Chen hasta Codd con ORACLE.[1]). En concreto, se han aplicado las siguientes reglas de transformación:

- RTECAR-1: Todos los tipos de entidad presentes en el esquema conceptual se transformarán en tablas o relaciones en el esquema relacional manteniendo el número y tipo de atributos, así como la característica de identificador de esos atributos.
- RTECAR-3.1: En las relaciones 1:N, la clave primaria de la entidad del lado 1 se convierte en una clave foránea en la tabla de la entidad del lado N.

Para cada tabla se mostrará la siguiente información:

- **Descripción.** Se describirá el origen de la tabla, indicando los elementos del modelo Entidad-Interrelación desde los que se ha obtenido.
- **Nombre de la tabla**
- **Atributos.** Se describirán los atributos que componen la tabla, distinguiendo su rol en cada caso con la siguiente notación:
 - Clave primaria
 - Clave ALTERNA, si existe
 - Claves **foráneas**, si existen
 - Resto de atributos
- **Esquema relacional.** Definición formal de la tabla, de acuerdo con el Modelo Relacional.

8.2.1. Tabla Usuario

- **Descripción:** la tabla Usuario se obtiene a partir de la entidad *Usuario*.
- **Nombre de la tabla:** usuario
- **Atributos:**
 - Clave primaria: id
 - Clave alterna: email
 - Claves foráneas: **id_rol**
 - Resto de atributos: nombre, apellidos, telefono, password, fecha_alta
- **Esquema relacional:** Usuario(id, nombre, apellidos, telefono, email, password, fecha_alta, id_rol)

8.2.2. Tabla Categoria

- **Descripción:** tabla derivada de la entidad *Categoría*.
- **Nombre de la tabla:** categoria
- **Atributos:**
 - Clave primaria: id
 - Resto de atributos: nombre, descripcion
- **Esquema relacional:** Categoria(id, nombre, descripcion)

8.2.3. Tabla Producto

- **Descripción:** procede de la entidad *Producto*.
- **Nombre de la tabla:** producto
- **Atributos:**
 - Clave primaria: id
 - Claves foráneas: **id_categoria**
 - Resto de atributos: nombre, descripcion, espacio, instalacion, precio, cantidad, imagen
- **Esquema relacional:** Producto(id, id_categoria, nombre, descripcion, espacio, instalacion, precio, cantidad, imagen)

8.2.4. Tabla Direccion

- **Descripción:** procede de la entidad *Dirección*.
- **Nombre de la tabla:** direccion
- **Atributos:**
 - Clave primaria: id
 - Claves foráneas: **id_usuario**
 - Resto de atributos: via, cod_postal, ciudad
- **Esquema relacional:** Direccion(id, id_usuario, via, cod_postal, ciudad)

8.2.5. Tabla Envio_servicio

- **Descripción:** procede de la entidad *Envio_Servicio*.
- **Nombre de la tabla:** envio_servicio
- **Atributos:**
 - Clave primaria: id
 - Claves foráneas: **id_direccion**
 - Resto de atributos: tiempo_envio, coste_base, activo
- **Esquema relacional:** Envio_servicio(id, tiempo_envio, coste_base, activo, id_direccion)

8.2.6. Tabla Contrato

- **Descripción:** procede de la entidad *Contrato*.
- **Nombre de la tabla:** contrato
- **Atributos:**
 - Clave primaria: id
 - Claves foráneas: **id_usuario, id_direccion**
 - Resto de atributos: fecha_inicio, fecha_fin, descripcion
- **Esquema relacional:** Contrato(id, id_usuario, id_direccion, fecha_inicio, fecha_fin, descripcion)

8.2.7. Tabla Contrato_productos

- **Descripción:** tabla procedente de la interrelación N:N entre *Contrato* y *Producto*.
- **Nombre de la tabla:** contrato_productos
- **Atributos:**
 - Clave primaria: id
 - Claves foráneas: **id_producto, id_contrato**
 - Resto de atributos: mensualidad
- **Esquema relacional:** Contrato_productos(id, id_producto, id_contrato, mensualidad)

8.2.8. Tabla Tipo

- **Descripción:** procede de la entidad *Tipo*.
- **Nombre de la tabla:** tipo
- **Atributos:**
 - Clave primaria: id
 - Resto de atributos: nombre, precio, descripcion, imagen
- **Esquema relacional:** Tipo(id, nombre, precio, descripcion, imagen)

8.2.9. Tabla Subscripcion

- **Descripción:** procede de la entidad *Subscripción*.
- **Nombre de la tabla:** subscripcion
- **Atributos:**
 - Clave primaria: id
 - Claves foráneas: **id_tipo, id_contrato**
 - Resto de atributos: mensualidad
- **Esquema relacional:** Subscripcion(id, id_tipo, id_contrato, mensualidad)

8.3. Normalización del modelo

La normalización del modelo descrito en la sección anterior pretende detectar y corregir redundancias e inconsistencias en la información representada, para lo cual se aplicarán las medidas correctoras que garanticen que las tablas obtenidas cumplen las siguientes formas normalizadas [1].

- Primera Forma Normal (FN1): una relación R satisface la FN1 si y sólo si, todos los dominios subyacentes de la relación R (atributos) contienen valores atómicos.
- Segunda Forma Normal (FN2): una relación R satisface la FN2 si y sólo si, satisface la FN1 y cada atributo de la relación depende funcionalmente de forma completa de la clave primaria de esa relación.
- Tercera Forma Normal (FN3): una relación R satisface la FN3 si y sólo si, satisface la FN2 y cada atributo no primo de la relación no depende funcionalmente de forma transitiva de la clave primaria de esa relación, es decir, no pueden existir dependencias transitivas entre los atributos que no forman parte de la clave primaria de esa relación R.
- Forma Normal de Boyce-Codd (FNBC): una relación R satisface la FNBC si y sólo si, se encuentra en FN1 y cada determinante funcional es una clave candidata de la relación R, denominándose determinante funcional a uno o un conjunto de atributos de una relación R del cual depende funcionalmente de forma completa algún otro atributo de la misma relación.

Todas las tablas definidas se encuentran en la Primera Forma Normal, puesto que en ninguna de ellas existen atributos múltiples.

8.3.1. Tabla Usuario

- **Claves candidatas:** id
- **Normalización:** La tabla Usuario presenta una dependencia funcional formada por la clave primaria y el resto de atributos. La tabla se encuentra en FNBC: el único determinante funcional es la clave primaria, por lo que la dependencia funcional con el resto de atributos es completa.

Usuario
id
nombre
apellidos
teléfono
email
password
fecha_alta
id_rol

Figura 8.1: Tabla Usuario en FNBC

8.3.2. Tabla Rol

- **Claves candidatas:** id.
- **Normalización:** La tabla posee solo dos atributos y una dependencia funcional $\text{id} \rightarrow \text{nombre}$. Cumple 1FN, 2FN, 3FN y FNBC.

Rol
id
nombre

Figura 8.2: Tabla Rol en FNBC

8.3.3. Tabla Categoría

- **Claves candidatas:** id.
- **Normalización:** Todos los atributos dependen completamente de la clave id, sin dependencias transitivas. La tabla se encuentra en FNBC.

Categorías
id
Nombre
Padre_id

Figura 8.3: Tabla Categoría en FNBC

8.3.4. Tabla Producto

- **Claves candidatas:** id.
- **Normalización:** Los atributos dependen totalmente de id. La clave foránea i_categoria no genera dependencias transitivas dentro de la tabla. La tabla se encuentra en FNBC.

Producto
id
i_categoria
nombre
descripcion
espacio
instalacion
precio
cantidad
imagen

Figura 8.4: Tabla Producto en FNBC

8.3.5. Tabla Direccion

- **Claves candidatas:** id.
- **Normalización:** Todos los atributos dependen directamente de id. id_usuario es clave foránea y no genera nuevas dependencias internas. La tabla está en FNBC.

Dirección
id
Id_Usuario
Via
Cod_Postal
Ciudad

Figura 8.5: Tabla Direccion en FNBC

8.3.6. Tabla Envio_servicio

- **Claves candidatas:** id.
- **Normalización:** Todos los atributos dependen totalmente de id. No hay atributos derivados ni dependencias transitivas. La tabla se encuentra en FNBC.

Envio_Servicio
id
Nombre
Tiempo_envio
Coste_Base
Activo
Id_Direccion

Figura 8.6: Tabla Envio_Servicio en FNBC

8.3.7. Tabla Contrato

- **Claves candidatas:** id.
- **Normalización:** No existen dependencias parciales ni transitivas. id_usuario e id_direccion son claves foráneas y no afectan a la normalización interna. La tabla se encuentra en FNBC.

Contrato
id
Id_Usuario
Id_Direccion
Fecha_Inicio
Fecha_Fin
Descripcion

Figura 8.7: Tabla Contrato en FNBC

8.3.8. Tabla Contrato_productos

- **Claves candidatas:** id.
- **Normalización:** Todos los atributos dependen completamente de id. id_producto e id_contrato actúan como claves foráneas. No existen dependencias transitivas. La tabla está en FNBC.

contrato_productos
id
id_producto
Id_Contrato
Mensualidad

Figura 8.8: Tabla Contrato_productos en FNBC

8.3.9. Tabla Tipo

- **Claves candidatas:** id.
- **Normalización:** Todos los atributos se determinan por la clave id. La tabla se encuentra en FNBC.

tipo
id
Nombre
Precio
Descripcion
Imagen

Figura 8.9: Tabla Tipo en FNBC

8.3.10. Tabla Subscripcion

- **Claves candidatas:** id.
- **Normalización:** Los atributos dependen totalmente de id. id_tipo e id_contrato son claves foráneas y no generan dependencias internas. La tabla cumple la FNBC.

Subscripcion
id
Id_Tipo
Id_Contrato
Mensualidad

Figura 8.10: Tabla Subscripcion en FNBC

8.4. Esquema relacional

Tabla	Atributos
Usuario	(id , nombre, apellidos, telefono, email, password, fecha_alta, id_rol)
Rol	(id , nombre)
Direccion	(id , id_usuario, via, cod_postal, ciudad)
Envio_Servicio	(id , nombre, tiempo_envio, coste_base, activo, id_direccion)
Contrato	(id , id_usuario, id_direccion, fecha_inicio, fecha_fin, descripcion)
Subscripcion	(id , id_tipo, id_contrato, mensualidad)
Tipo	(id , nombre, precio, descripcion, imagen)
Producto	(id , id_categoria, nombre, descripcion, espacio, instalacion, precio, cantidad, imagen)
Categorias	(id , nombre, padre_id)
Contrato_productos	(id , id_producto, id_contrato, mensualidad)

8.5. Diagrama relacional

El diagrama relacional del modelo propuesto se muestra en la figura 8.11.

Figura 8.11: Diagrama relacional

Capítulo 9

Diseño arquitectónico

9.1. Introducción

9.2. Diagrama de despliegue

9.2.1. Descripción de los nodos

9.2.2. Descripción de los componentes

Capítulo 10

Diseño de la interfaz

10.1. Introducción

10.2. Características comunes

10.3. Interfaz del módulo Usuario...

10.3.1. Descripción general

hjk

10.3.2. Descripción detallada

hjk

10.3.3. Otros elementos de la interfaz

hjk

10.4. Interfaz del módulo Administrador

10.4.1. Descripción general

hjk

10.4.2. Descripción detallada

hjk

10.4.3. Otros elementos de la interfaz

hjk

Parte V

Pruebas

Capítulo 11

Pruebas

11.1. Introducción

11.2. Prueba de usuarios públicos

11.2.1. Pruebas CU-1.1

- Prueba realizada
- Resultado de la prueba
- Prueba realizada
- Resultado de la prueba

Parte VI

Conclusiones y futuras mejoras

Capítulo 12

Conclusiones

12.1. Introducción

En este capítulo se exponen la conclusiones obtenidas sobre el desarrollo del proyecto, y en relación con los objetivos planteados inicialmente en el capítulo 2 frente al resultado final alcanzado tras la finalización del mismo y las pruebas realizadas.

Con carácter general, el objetivo principal, consistente en el desarrollo de un sistema informático ...

12.2. Conclusiones específicas

12.3. Conclusiones personales y profesionales

A nivel personal, el presente proyecto nos ha permitido reciclar, actualizar ...

En cuanto al ámbito profesional, se han logrado los objetivos propuestos:

■

Capítulo 13

Futuras mejoras

13.1. Introducción

El sistema desarrollado en este proyecto es susceptible de mejoras y ampliaciones, como cualquier otro software. De hecho, se identificaron muchas oportunidades durante el desarrollo.

En las siguientes secciones se desglosan las mejoras y ampliaciones propuestas, según cada una de las categoría expuestas.

13.2. Mejoras y nuevas funciones

Basándonos en lo que se ha desarrollado hasta ahora, se identifican una serie de funciones adicionales y mejoras que se pueden realizar. Estas mejoras se describen a continuación, divididas por tipo de usuario y mejoras generales.

13.2.1. Mejoras para usuarios

- G

13.2.2. Mejoras para administrador

-

13.2.3. Mejoras de funcionamiento

-

13.2.4. Integración con otras aplicaciones

13.2.5. Exportabilidad

Capítulo 14

ANEXOS

Este apartado está destinado a incluir todo aquello que queráis añadir de manera adicional al trabajo.

Bibliografía

- [1] . []
- [2] Microsoft. *Visual Studio Code*. [En línea. Última consulta:] URL: <https://code.visualstudio.com/>.
- [3] World Wide Web Consortium. *World Wide Web Consortium*. [En Línea. Última consulta:] URL: <https://www.w3.org/>.