

DUIT

R1L2

Diagrama E/R

Autores:

Aleixo Fernández Cuevas

Cristo Manuel Navarro Martín

Nombre del fichero:

DAW_PRW_R1L2_UT01.3 – Diagrama E/R

Fecha de esta versión:

01/02/2026

Ciclo Formativo:

Desarrollo de Aplicaciones Web – Semipresencial (DAWN)

Módulo:

Proyecto de Desarrollo de Aplicaciones Web (PRW)

Historial de revisiones

Fecha	Descripción	Autores
21/12/2025	Fase de análisis. Versión inicial	Aleixo F. Cuevas / Cristo N. Martín
10/01/2026	Primera corrección	Aleixo F. Cuevas / Cristo N. Martín
01/02/2026	Segunda corrección	Aleixo F. Cuevas / Cristo N. Martín

Índice

1. Resumen Ejecutivo.....	5
2. Objetivo del Modelo de Datos.....	5
3. Identificación de Entidades.....	5
3.1 Entidades Principales.....	5
3.2 Entidades Secundarias	5
3.3 Consideraciones Generales.....	6
4. Descripción del Diagrama Relacional.....	6
6. Justificación del Diseño.....	7
7. Especificación Detallada de Entidades.....	8
7.1 Resumen de Entidades del Sistema.....	8
7.2. AppUser (Usuario de la Aplicación).....	9
7.3. UserRole (Rol de Usuario).....	9
7.4. Address (Dirección).....	10
7.5. ProfessionalProfile (Perfil Profesional).....	10
7.6. Category (Categoría).....	10
7.7. ServiceRequest (Solicitud).....	10
7.8. JobApplication (Postulación).....	11
7.9. ServiceJob (Trabajo).....	11
7.10 Rating (Valoración).....	11
7.11 AccessLog (Registro de Accesos).....	11
8. Relación entre Tablas Intermedias.....	12
8.1. professional_category (Many-to-Many).....	12
9. Reglas de Negocio e Integridad Referencial.....	12
9.1. Usuarios y Roles.....	12
9.2. Solicitudes de Servicio.....	12
9.3. Postulaciones.....	12
9.4. Trabajos.....	12
9.5. Valoraciones.....	13

10. Consideraciones Técnicas de Implementación.....	13
10.1. Seguridad	13
10.2. Optimización y Rendimiento	13
11. Notas Adicionales y Consideraciones Futuras	14
11.1. Características Implementadas.....	14
11.2. Consideraciones de Escalabilidad	14
11.3. Integridad y Coherencia del Sistema.....	14
12. Reparto de tareas y responsabilidades	15

1. Resumen Ejecutivo

El modelo se ha realizado a partir de los requisitos definidos en las fases anteriores del proyecto y del diagrama de casos de uso desarrollado en la UT01.2.

El objetivo principal de este documento es mostrar cómo se organiza la información del sistema, definiendo las entidades, sus atributos y las relaciones entre ellas, de forma que sirva como base para la implementación de la base de datos del proyecto.

2. Objetivo del Modelo de Datos

El objetivo del modelo de datos es definir cómo se va a almacenar y relacionar la información que utilizará la aplicación. Este diseño permite gestionar usuarios con distintos roles, crear y seguir solicitudes de trabajo, asignar trabajos a los profesionales y realizar valoraciones una vez finalizado un servicio, asegurando que los datos se mantengan organizados y sin duplicidades.

3. Identificación de Entidades

Tras analizar el funcionamiento del sistema y los requisitos definidos, se han identificado las entidades necesarias para representar la información y los procesos principales de la aplicación.

3.1 Entidades Principales

Se consideran entidades principales aquellas que representan los elementos clave del funcionamiento del sistema y sobre las que se articulan los principales procesos de la aplicación.

- **AppUser (Usuario):** Representa a cualquier persona que utiliza el sistema, independientemente de su rol (cliente, profesional, administrador o moderador).
- **ServiceRequest (Solicitud de Servicio):** Representa una petición de servicio creada por un cliente, que puede recibir postulaciones por parte de profesionales.
- **JobApplication (Postulación):** Representa la propuesta realizada por un profesional para llevar a cabo una solicitud concreta, actuando como paso intermedio entre la solicitud y el trabajo.
- **ServiceJob (Trabajo):** Representa el servicio que se realiza una vez que una postulación ha sido aceptada.

3.2 Entidades Secundarias

Se consideran entidades secundarias aquellas que complementan a las entidades principales del sistema, aportando información adicional necesaria para su funcionamiento. Estas entidades no representan procesos principales por sí mismas, pero permiten ampliar y organizar los datos de forma estructurada.

Por ejemplo, entidades como **UserRole**, **Category** o **Address** permiten clasificar a los usuarios, agrupar los servicios disponibles y asociar información de localización, respectivamente. Otras entidades como **ProfessionalProfile**, **Rating** y **AccessLog** complementan el sistema con información específica de perfiles profesionales, valoraciones y auditoría de accesos.

3.3 Consideraciones Generales

Cada entidad cuenta con una clave primaria que permite identificar de forma única cada registro dentro del sistema.

Además, las relaciones entre entidades se gestionan mediante claves foráneas, garantizando la integridad referencial de los datos y la coherencia del modelo.

Todas las entidades principales heredan de BaseEntity, que proporciona campos de auditoría automática (created_by, created_at, updated_by, updated_at), permitiendo un seguimiento completo de los cambios realizados en el sistema.

4. Descripción del Diagrama Relacional

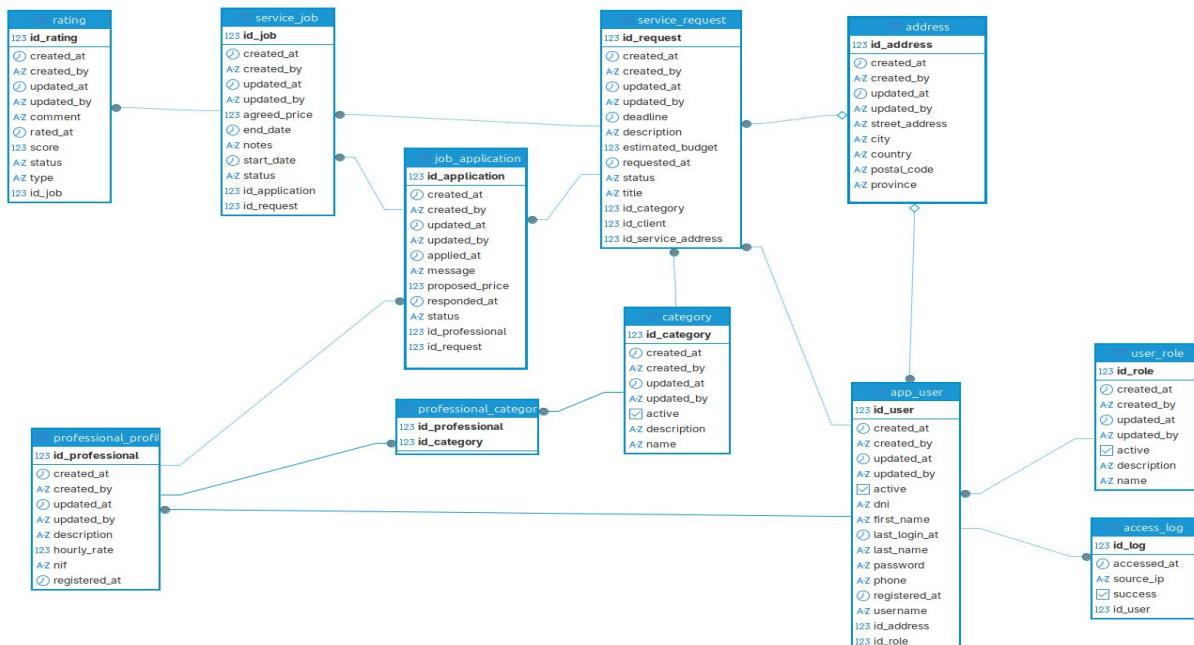
El modelo de datos del sistema se ha representado mediante un diagrama relacional utilizando la notación Crow's Foot, en el que se muestran las entidades identificadas, sus atributos principales, las claves primarias y las claves foráneas que definen las relaciones entre ellas.

El diagrama refleja la estructura de la base de datos diseñada para la aplicación, permitiendo visualizar de forma clara las relaciones y cardinalidades entre las distintas entidades, así como la dependencia existente entre los procesos principales del sistema.

En el modelo se distingue una entidad central, **AppUser**, a partir de la cual se articulan el resto de entidades del sistema. A través de esta entidad se gestionan los distintos roles, las direcciones asociadas y, en el caso de los profesionales, la información específica almacenada en el perfil profesional.

Asimismo, el diagrama representa el flujo completo del proceso principal de la aplicación, desde la creación de una solicitud por parte de un cliente, pasando por las postulaciones realizadas por los profesionales, hasta la generación del trabajo una vez aceptada una postulación y la posterior valoración entre las partes implicadas.

Este diagrama sirve como referencia directa para la implementación de la base de datos en **PostgreSQL**, garantizando la coherencia entre el diseño conceptual y la estructura física de las tablas, así como la correcta aplicación de las restricciones de integridad definidas.



6. Justificación del Diseño

Se ha optado por una única entidad **AppUser** como núcleo del modelo, concentrando en ella la información común a todos los perfiles del sistema. La diferenciación de permisos y comportamientos se realiza mediante la entidad **UserRole**, lo que permite gestionar distintos tipos de usuarios sin duplicar estructuras ni datos. Esta decisión simplifica el mantenimiento del sistema, mejora la consistencia de la información y facilita la incorporación de nuevos roles en el futuro.

La separación entre **ServiceRequest**, **JobApplication** y **ServiceJob** responde a la necesidad de representar correctamente el ciclo de vida de los servicios ofrecidos en la plataforma. Una solicitud puede recibir múltiples postulaciones por parte de profesionales, pero no todas ellas derivan en un trabajo. El trabajo únicamente se genera cuando una postulación es aceptada, lo que permite reflejar de forma realista el proceso de selección y ejecución del servicio, evitando un modelo rígido y poco flexible.

La entidad **ProfessionalProfile** se ha diseñado para almacenar información específica de los usuarios que ofrecen servicios, como la descripción profesional, el precio por hora o el NIF. De este modo, se evita incluir campos innecesarios en la entidad **AppUser** para aquellos perfiles que no desempeñan actividades profesionales, manteniendo un diseño más limpio y modular.

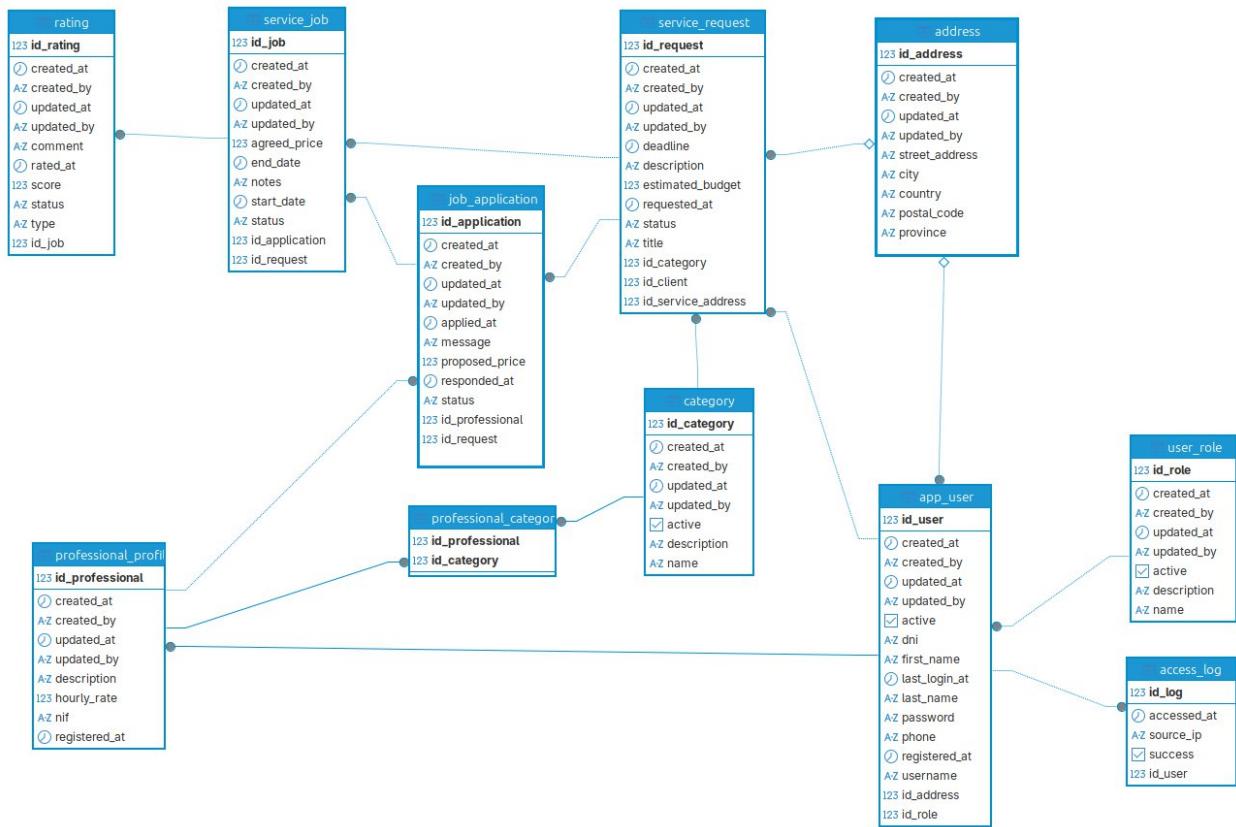
La entidad **Address** se ha separado de la entidad **AppUser** para permitir un registro progresivo de los datos y mejorar la flexibilidad del sistema. La dirección se asocia de forma opcional a cada usuario, lo que permite que el registro inicial sea sencillo y que la información de localización se complete posteriormente cuando sea necesario. Además, esta separación permite reutilizar direcciones para las solicitudes de servicio, donde se indica el lugar donde se realizará el trabajo.

Por último, la entidad **Rating** garantiza que las opiniones entre clientes y profesionales estén siempre vinculadas a trabajos reales y finalizados. Esto refuerza la integridad de los datos y la fiabilidad del sistema, evitando valoraciones arbitrarias y asegurando un sistema de reputación coherente. El sistema permite valoraciones bidireccionales, donde tanto el cliente como el profesional pueden evaluar su experiencia mutua.

Además, la entidad **AccessLog** proporciona trazabilidad completa de los accesos al sistema, registrando información relevante como la IP de origen, fecha y hora, y éxito o fallo del intento de acceso, lo que contribuye a la seguridad y auditoría del sistema.

7. Especificación Detallada de Entidades

El presente apartado describe de forma detallada el **modelo de datos del sistema Duit**, compuesto por **10 entidades principales**.



7.1 Resumen de Entidades del Sistema

Nº	Entidad	Tipo	Descripción
1	AppUser	Núcleo	Usuarios del sistema
2	UserRole	Núcleo	Roles y permisos
3	Address	Núcleo	Direcciones
4	ProfessionalProfile	Núcleo	Perfil profesional
5	Category	Núcleo	Categorías de servicios
6	ServiceRequest	Flujo	Solicitudes de servicio
7	JobApplication	Flujo	Postulaciones
8	ServiceJob	Flujo	Trabajos asignados
9	Rating	Soporte	Valoraciones
10	AccessLog	Auditoría	Registro de accesos

7.2. AppUser (Usuario de la Aplicación)

Campo	Tipo	Restricciones	Descripción
id_user	Long	PK, Auto	Identificador del usuario
first_name	String(100)	NOT NULL	Nombre
last_name	String(150)	NULL	Apellidos
dni	String(9)	UNIQUE	DNI
username	String(100)	NOT NULL, UNIQUE	Email
password	String(255)	NOT NULL	Contraseña cifrada
phone	String(20)	NULL	Teléfono
active	Boolean	DEFAULT TRUE	Estado
registered_at	LocalDateTime	NULL	Registro
last_login_at	LocalDateTime	NULL	Último acceso
id_role	Long	FK	Rol
id_address	Long	FK	Dirección
created_at	LocalDateTime	NULL	Creación
updated_at	LocalDateTime	NULL	Actualización

Relaciones

Tipo	Entidad
ManyToOne	UserRole
ManyToOne	Address
OneToOne	ProfessionalProfile
OneToMany	ServiceRequest
OneToMany	AccessLog

7.3. UserRole (Rol de Usuario)

Campo	Tipo	Restricciones	Descripción
id_role	Long	PK	Identificador
name	ENUM	UNIQUE	Rol
description	String(100)	NULL	Descripción
active	Boolean	DEFAULT TRUE	Estado

Valores ENUM

Rol	Descripción
ADMIN	Administrador
USER	Cliente
PROFESSIONAL	Profesional

7.4. Address (Dirección)

Campo	Tipo	Restricciones	Descripción
id_address	Long	PK	Identificador
street_address	String(200)	NOT NULL	Calle
city	String(100)	NOT NULL	Ciudad
postal_code	String(5)	NULL	CP
province	String(100)	NOT NULL	Provincia
country	String(50)	DEFAULT España	País

7.5. ProfessionalProfile (Perfil Profesional)

Campo	Tipo	Restricciones	Descripción
id_professional	Long	PK/FK	Usuario
description	TEXT	NOT NULL	Descripción
hourly_rate	Decimal(8,2)	NOT NULL	€/hora
nif	String(9)	UNIQUE	NIF

7.6. Category (Categoría)

Campo	Tipo	Restricciones	Descripción
id_category	Long	PK	Identificador
name	String(100)	UNIQUE	Nombre
description	String(200)	NULL	Descripción
active	Boolean	DEFAULT TRUE	Estado

7.7. ServiceRequest (Solicitud)

Campo	Tipo	Restricciones	Descripción
id_request	Long	PK	Identificador
title	String(150)	NOT NULL	Título
description	TEXT	NOT NULL	Descripción
status	ENUM	NOT NULL	Estado
id_client	Long	FK	Cliente
id_category	Long	FK	Categoría
id_service_address	Long	FK	Dirección

Estados

Estado	Descripción
DRAFT	Borrador
PUBLISHED	Publicada
IN_PROGRESS	En progreso
COMPLETED	Completada
CANCELLED	Cancelada

7.8. JobApplication (Postulación)

Campo	Tipo	Restricciones	Descripción
id_application	Long	PK	Identificador
id_request	Long	FK	Solicitud
id_professional	Long	FK	Profesional
status	ENUM	NOT NULL	Estado

7.9. ServiceJob (Trabajo)

Campo	Tipo	Restricciones	Descripción
id_job	Long	PK	Identificador
agreed_price	Decimal(8,2)	NOT NULL	Precio
status	ENUM	NOT NULL	Estado

7.10 Rating (Valoración)

Campo	Tipo	Descripción
id_rating	Long	Identificador
score	Integer	1–5
type	ENUM	Tipo
status	ENUM	Estado

7.11 AccessLog (Registro de Accesos)

Campo	Tipo	Descripción
id_log	Long	Identificador
accessed_at	LocalDateTime	Fecha
source_ip	String(45)	IP
success	Boolean	Resultado

8. Relación entre Tablas Intermedias

8.1. professional_category (Many-to-Many)

Descripción:

Tabla intermedia que gestiona la relación **muchos a muchos** entre **ProfessionalProfile** y **Category**, permitiendo que un profesional pueda ofrecer servicios en múltiples categorías y que una categoría tenga asociados múltiples profesionales.

Relación representada:

- ProfessionalProfile ↔ Category

9. Reglas de Negocio e Integridad Referencial

Este apartado define las **reglas de negocio principales** que garantizan la coherencia funcional del sistema y la **integridad referencial** del modelo de datos.

9.1. Usuarios y Roles

- Todo usuario debe tener **un rol asignado**
- Los usuarios con rol PROFESSIONAL deben disponer obligatoriamente de un **ProfessionalProfile**
- El **DNI** y el **username (email)** deben ser **únicos** en el sistema
- Un usuario desactivado no puede iniciar sesión ni operar en el sistema

9.2. Solicitudes de Servicio

- Toda solicitud debe tener:
 - o Un **cliente** asociado
 - o Una **categoría** asignada
- Una solicitud puede recibir **múltiples postulaciones**
- Solo se puede crear un **ServiceJob** cuando una postulación es aceptada
- Una solicitud no puede estar en más de un estado activo simultáneamente

9.3. Postulaciones

- Un profesional puede postularse a **múltiples solicitudes**
- No se permite más de una postulación del mismo profesional a la **misma solicitud**
- El mensaje de presentación y el precio propuesto son **opcionales**
- Una postulación solo puede ser aceptada si la solicitud está en estado PUBLISHED

9.4. Trabajos

- Un trabajo solo se crea cuando se acepta una postulación
- Cada trabajo está vinculado:
 - o A una **solicitud**
 - o A una **postulación específica**
- El **precio acordado** es obligatorio
- Un trabajo debe pasar por un flujo de estados válido (CREATED → IN_PROGRESS → COMPLETED / CANCELLED)

9.5. Valoraciones

- Las valoraciones son **bidireccionales**:
 - Cliente → Profesional
 - Profesional → Cliente
- Solo se pueden valorar **trabajos completados**
- La puntuación debe estar comprendida entre **1 y 5**
- Las valoraciones pueden quedar en estado **pendiente de moderación**

10. Consideraciones Técnicas de Implementación

10.1. Seguridad

- **Autenticación:**
Implementada mediante **Spring Security**, utilizando **BCrypt** para el cifrado de contraseñas.
- **Autorización:**
Sistema basado en roles (ADMIN, USER, PROFESSIONAL), controlado mediante anotaciones de seguridad.
- **Auditoría:**
Uso de **JPA Auditing** con @EntityListeners y anotaciones:
 - @CreatedBy
 - @CreatedDate
 - @LastModifiedBy
 - @LastModifiedDate
- **Validación:**
Validaciones a nivel de entidad mediante **Jakarta Validation** (@NotNull, @Size, @Email, etc.).

10.2. Optimización y Rendimiento

- **Índices:**
Definidos en campos de consulta frecuente para mejorar el rendimiento.
- **Lazy Loading:**
Configurado en las relaciones para evitar el problema **N+1**.
- **Soft Delete:**
Implementado mediante el campo `active` en entidades principales, evitando borrados físicos.
- **Lombok:**
Utilizado para reducir código boilerplate y mejorar la mantenibilidad del proyecto.

11. Notas Adicionales y Consideraciones Futuras

Este apartado recoge observaciones complementarias sobre el diseño del sistema, así como posibles líneas de evolución y mejora del proyecto.

11.1. Características Implementadas

El sistema incorpora una serie de características técnicas orientadas a garantizar la robustez, mantenibilidad y coherencia del modelo de datos:

- Sistema de **auditoría completa** mediante herencia de una entidad base (BaseEntity), que centraliza los campos de trazabilidad.
- Gestión de estados mediante **enumerados Java (ENUM)**, asegurando consistencia y control del ciclo de vida de las entidades.
- **Validaciones exhaustivas** a nivel de entidad mediante **Jakarta Validation**, reforzando la integridad de los datos.
- Uso de **relaciones bidireccionales**, con gestión automática del ciclo de vida según las necesidades funcionales.
- Soporte para **direcciones reutilizables**, permitiendo su asociación tanto a usuarios como a solicitudes de servicio.

11.2. Consideraciones de Escalabilidad

El diseño del sistema ha sido planteado teniendo en cuenta su posible crecimiento y evolución futura:

- La arquitectura modular permite la incorporación de **nuevos roles de usuario** sin necesidad de modificar la estructura base del sistema.
- La separación clara entre entidades del núcleo, flujo de trabajo y soporte facilita una futura **migración hacia una arquitectura de microservicios**.
- El uso de **índices simples y compuestos** optimiza consultas complejas y mejora el rendimiento en escenarios de alta carga.
- La estructura del modelo permite la incorporación de **mecanismos de caché** (por ejemplo, Redis) de forma sencilla y no intrusiva.

11.3. Integridad y Coherencia del Sistema

Para garantizar la coherencia global del sistema, se han aplicado las siguientes medidas:

- Todas las relaciones incluyen **restricciones de integridad referencial**, evitando inconsistencias en la base de datos.
- Los campos obligatorios están claramente definidos mediante restricciones NOT NULL.
- Los campos únicos previenen la duplicación de información crítica a nivel de base de datos.
- Las operaciones en cascada están configuradas de forma selectiva, atendiendo a los requisitos funcionales de cada relación.

El diseño se encuentra **alineado con los requisitos UT01.2 y con los casos de uso definidos en secciones previas**, garantizando la trazabilidad entre análisis y diseño.

12. Reparto de tareas y responsabilidades

El desarrollo del modelo de datos, la especificación de entidades y la definición de las reglas de negocio del sistema **Duit** se ha llevado a cabo de forma colaborativa por los integrantes del grupo, estableciendo un reparto claro de tareas y responsabilidades.

Cristo Manuel Navarro Martín ha asumido principalmente el diseño del modelo de datos, la identificación y definición de las entidades del sistema, así como la especificación de sus relaciones, restricciones de integridad y consideraciones de escalabilidad.

Aleixo Fernández Cuevas ha participado en la definición de las reglas de negocio, la documentación técnica asociada a las entidades y la revisión global del diseño para asegurar la coherencia entre el modelo de datos, los casos de uso y los requisitos funcionales.

Las decisiones finales relativas a la estructura del modelo, las relaciones entre entidades y las reglas de integridad han sido consensuadas entre ambos miembros del grupo, garantizando un diseño consistente, alineado con los objetivos del sistema y con las necesidades funcionales definidas previamente.