

PROYECTO INTERMODULAR - 2º DAW 25/26

0. ÍNDICE

1. Portada
2. Resumen del proyecto
3. Introducción
 - 3.1 Contexto y justificación del proyecto
 - 3.2 Objetivos del proyecto
 - 3.2.1 Objetivo general
 - 3.2.2 Objetivos específicos
 - 3.3 Alcance y limitaciones del trabajo
4. Descripción General
5. Funcionalidades Principales
6. Caracterización de empresas del sector
 - 6.1 Zendesk
 - 6.2 Freshdesk
 - 6.3 Zoho Desk
 - 6.4 Justificación de la empresa seleccionada
7. Relación con los ODS
 - 7.1 ODS 8 – Trabajo decente y crecimiento económico
 - 7.2 ODS 9 – Industria, innovación e infraestructura
 - 7.3 ODS 12 – Producción y consumo responsables
 - 7.4 ODS 17 – Alianzas para lograr los objetivos
 - 7.5 Justificación de la relación entre los ODS y la actividad
8. Identificación de riesgos laborales
 - 8.1 Riesgos ergonómicos
 - 8.2 Riesgos derivados del uso de pantallas
 - 8.3 Riesgos psicosociales
 - 8.4 Riesgos eléctricos e infraestructura
 - 8.5 Riesgos de seguridad informática
 - 8.6 Conclusiones del análisis
9. Tecnologías y Herramientas
10. Público Objetivo
11. Modelo de Monetización

12. Desarrollo del proyecto (MVP y fases)

- 12.1 Definición del MVP
- 12.2 Diseño de arquitectura y base de datos
 - 12.2.1 Arquitectura general
 - 12.2.2 Modelo de Datos
 - 12.2.2.1 Entidad Empresa
 - 12.2.2.2 Entidad Usuario
 - 12.2.2.3 Entidad Ticket
 - 12.2.2.4 Entidad Mensaje
 - 12.2.2.5 Entidad RespuestaAutomática
 - 12.2.2.6 Entidad Métrica
 - 12.3 Implementación de la versión inicial
 - 12.4 Pruebas con usuarios reales y recogida de feedback
 - 12.4.1 Casos de prueba funcionales
 - 12.4.2 Pruebas con usuarios reales
 - 12.4.3 Casos de feedback y mejora
 - 12.4.4 Mejora continua

13. Casos de Uso

14. Diseño de funcionalidades principales

15. Tecnologías y justificación técnica

16. Seguridad y privacidad

17. Escalabilidad y mejoras futuras

18. Modelo de negocio

19. Plan de desarrollo

20. Resultados y Análisis

- 20.1 Funcionamiento del sistema
- 20.2 Usabilidad y experiencia de usuario
- 20.3 Rendimiento y estabilidad

21. Conclusiones y Recomendaciones

- 21.1 Conclusiones
- 21.2 Recomendaciones para futuros proyectos

22. Bibliografía

23. Anexos

- 23.1 Documentación técnica
- 23.2 Ficha de autoevaluación
- 23.3 Ficha de coevaluación entre miembros del equipo

1. Portada



Nombre proyecto: SupportStation (SaaS de Atención al Cliente)

Centro educativo: IES Dos Mares

Fecha de presentación: 18-02-2026

Módulos profesionales implicados:

- Desarrollo Web en Entorno Servidor
- Desarrollo Web en Entorno Cliente
- Despliegue de Aplicaciones Web
- Empresa e Iniciativa Emprendedora

Profesor responsable: Raúl Almarcha Olivares

Equipo: Rubén Martínez Agramunt y Alejandro Tarazona Sáez.

The screenshot displays a customer service platform's user interface. On the left, there are filters for 'Assignee' (set to 'You'), 'Followers' (searchable), 'Form' (set to 'Support Form'), 'Tags' (including 'vip' and 'priority_support'), and 'Priority' (set to 'High'). The main area shows a conversation with 'Riley Green via Whatsapp' (Active). The conversation history includes a message from 'Alex via Whatsapp' confirming account upgrade and a message from 'Riley Green via Whatsapp' asking for an email receipt. Below the messages is a dropdown menu with options: 'Call', 'Whatsapp' (disabled), 'Email' (selected and highlighted in yellow), and 'Internal note'. The right side of the interface shows 'Interactions' with a list of recent activities: 'Conversation with Riley Green' (Active now), 'Account upgrade' (Friday 12:08 * Shopify), 'Campaign opened' (Thursday 10:32 * Mailchimp), 'Issue linked to project' (Wednesday 9:08 * Jira), 'Streaming issue' (Wednesday 9:04), and 'Priority migration'. Each interaction is accompanied by a small circular icon representing the system it originated from.

2. Resumen del proyecto

Después de analizar distintas ideas y valorar múltiples propuestas, hemos identificado una iniciativa sólida con alto potencial de desarrollo: la creación de **SupportStation**, un **SaaS (Software as a Service)** orientado a la atención al cliente.

El objetivo principal de la plataforma es **optimizar la comunicación entre las empresas y sus usuarios finales**, centralizando, automatizando y mejorando la gestión de consultas, incidencias y solicitudes. Esto permitirá ofrecer una experiencia de usuario **más rápida, eficiente y personalizada**.

La plataforma está diseñada para adaptarse a **distintos tipos de negocios**, independientemente de su tamaño, incorporando herramientas modernas y escalables que faciliten la fidelización del cliente y la mejora continua del servicio.

Entre los aspectos clave que se tendrán en cuenta durante el desarrollo del proyecto destacan:

- **Funcionalidad:** ofrecer un conjunto de herramientas esenciales para la atención al cliente, incluyendo tickets, chat y métricas.
- **Usabilidad:** garantizar una interfaz clara, intuitiva y accesible para todo tipo de usuarios.
- **Valor añadido:** proporcionar una solución que ayude a las PYMEs a mejorar la eficiencia del soporte y la satisfacción del cliente.
- **Potencial de monetización:** **alto**, mediante un modelo de suscripción mensual adaptable a diferentes necesidades.
- **Dificultad de creación:** **media-alta**, debido a la necesidad de integrar distintas funcionalidades y canales de comunicación.
- **Tiempo de puesta en marcha:** estimado entre 1 y 4 semanas para un **MVP (Producto Mínimo Viable)** que permita validar la idea y su viabilidad.

Además, el proyecto se gestionará de manera colaborativa, utilizando **herramientas de planificación y seguimiento** como Trello, asegurando un control eficiente del flujo de trabajo y la documentación técnica.

Enlace a Trello:

<https://trello.com/invite/b/69394f584ac1ebd90190358c/ATTI8f3504336911af3c82eab2387c01ce55C2E745A5/poyecto-intermodular>



3. Introducción

3.1 Contexto y justificación del proyecto

Las pymes suelen enfrentarse a varios problemas comunes:

- Falta de centralización: los mensajes llegan por email, WhatsApp, redes sociales, etc.
- Pérdida de información: no existe un registro histórico organizado.
- Tiempos de respuesta elevados: no hay herramientas que prioricen o automatizan tareas.
- Dificultad para medir el rendimiento: no se registran métricas de soporte.

El SaaS propuesto resuelve estos problemas mediante:

- Un panel único para gestionar todas las consultas.
- Un sistema de tickets con estados y asignación.
- Chat en tiempo real para mejorar la experiencia del cliente.
- Automatizaciones simples para reducir carga de trabajo.
- Panel de métricas para evaluar el rendimiento del equipo.

Potencial de monetización: Alto.

Dificultad de creación: Media-Alta.

Tiempo de desarrollo estimado: 1-4 semanas para un MVP.

3.2 Objetivos del proyecto

Objetivo general:

Desarrollar una plataforma SaaS de atención al cliente que permita a las empresas centralizar, automatizar y mejorar la gestión de la comunicación con sus usuarios finales, optimizando los tiempos de respuesta y la calidad del servicio ofrecido.

Objetivos específicos:

1. **Centralizar la comunicación con los clientes:** integrar todos los canales (correo electrónico, chat, redes sociales) en un único panel de control para facilitar la gestión de consultas e incidencias.
2. **Reducir los tiempos de respuesta:** implementar funcionalidades de automatización básica que agilicen la resolución de solicitudes repetitivas.
3. **Facilitar la gestión de tickets:** crear un sistema de asignación y seguimiento de incidencias que permita organizar y priorizar las solicitudes de los clientes.
4. **Proporcionar métricas y estadísticas:** ofrecer un panel de indicadores que permita evaluar el rendimiento del equipo de soporte y la satisfacción de los clientes.

5. **Diseñar una arquitectura escalable y segura:** garantizar que el sistema pueda evolucionar y adaptarse a nuevas funcionalidades, integraciones y un mayor número de usuarios.
6. **Validar la viabilidad del MVP:** desarrollar un Producto Mínimo Viable que permita comprobar la funcionalidad y aceptación del sistema antes de futuras ampliaciones.

3.3 Alcance y limitaciones del trabajo

Alcance del proyecto:

El proyecto **SupportStation** se centra en el desarrollo de un **Producto Mínimo Viable (MVP)** que incluya las funcionalidades esenciales para validar la idea y demostrar su viabilidad técnica y funcional. Dentro de este alcance se incluyen:

- **Registro y autenticación de usuarios** (empresas, agentes y clientes).
- **Gestión básica de tickets:** creación, asignación, seguimiento y cierre.
- **Chat en tiempo real** entre clientes y agentes.
- **Respuestas automáticas simples** a consultas frecuentes.
- **Panel básico de métricas** que permita visualizar información sobre tickets y tiempos de respuesta.

El desarrollo contempla tanto el **backend**, la **base de datos** y la **interfaz web frontend**, utilizando tecnologías como Django, React, PostgreSQL y Docker, con un enfoque colaborativo mediante Git y seguimiento en Trello.

Limitaciones del proyecto:

Debido a las restricciones de tiempo, recursos y al carácter académico del proyecto, se establecen las siguientes limitaciones:

1. **Funcionalidades avanzadas fuera del MVP:** integraciones con WhatsApp Business, redes sociales y automatizaciones complejas no se implementan en esta fase inicial.
2. **Sistema de facturación completo:** queda pendiente para futuras versiones.
3. **Escalabilidad y rendimiento en entornos de producción:** aunque la arquitectura permite crecimiento, no se realizará despliegue masivo ni pruebas de carga intensivas.

4. **Alcance temporal:** el proyecto se desarrolla en un marco de 4 semanas, por lo que algunas mejoras y refinamientos se posponen a iteraciones posteriores.
 5. **Pruebas limitadas con usuarios reales:** se realizarán pruebas con un número reducido de usuarios (compañeros o familiares) para evaluar usabilidad y funcionalidad, sin llegar a un entorno de clientes reales.
-

4. Descripción General

SupportStation es un **Software as a Service (SaaS)** diseñado para la **gestión y atención al cliente de empresas**, con un enfoque en eficiencia y experiencia del usuario.

La plataforma permite a las compañías **centralizar y automatizar la comunicación con sus clientes** a través de múltiples canales, como correo electrónico, chat en tiempo real y redes sociales. Esto facilita la gestión de consultas, incidencias y solicitudes, optimizando los tiempos de respuesta y mejorando la calidad del servicio.

Con un diseño **intuitivo y escalable**, la solución está pensada para adaptarse a empresas de distintos tamaños, contribuyendo a la **eficiencia del equipo de soporte** y a la **satisfacción del cliente final**, al ofrecer una experiencia de atención más rápida, organizada y profesional.

5. Funcionalidades Principales

1. **Gestión de tickets:** Centralización de todas las solicitudes de clientes en un solo panel.
 2. **Chat en tiempo real:** Comunicación instantánea con los clientes desde la web o la app.
 3. **Automatización:** Respuestas automáticas a consultas frecuentes y asignación de tickets a distintos agentes.
 4. **Integración multicanal:** Email, WhatsApp, redes sociales, etc.
 5. **Panel de métricas:** Seguimiento de tiempos de respuesta, satisfacción del cliente y volumen de tickets.
-

6. Caracterización de empresas del sector

Para contextualizar el proyecto SupportStation, se han analizado varias empresas representativas del sector de soluciones de atención al cliente basadas en software (SaaS). Estas empresas ofrecen plataformas que permiten centralizar la comunicación con los clientes, gestionar tickets y automatizar respuestas. Entre las más destacadas se encuentran:

Zendesk

- **Tipo de empresa:** Privada, multinacional.
- **Sector:** Software de atención al cliente y CRM.
- **Ubicación:** San Francisco, Estados Unidos.
- **Tamaño:** Más de 5.000 empleados.
- **Productos y servicios:** Plataforma de atención al cliente multicanal, tickets, chat en tiempo real, integraciones con redes sociales y correo electrónico.
- **Mercado objetivo:** Empresas de todos los tamaños, especialmente grandes y medianas.
- **Justificación:** Zendesk es líder en el mercado y sirve como referencia para funcionalidades avanzadas de soporte y escalabilidad.

Freshdesk

- **Tipo de empresa:** Privada, multinacional.
- **Sector:** SaaS de atención al cliente y helpdesk.
- **Ubicación:** San Francisco, Estados Unidos (con oficinas en India y Europa).
- **Tamaño:** Más de 4.000 empleados.
- **Productos y servicios:** Sistema de tickets, automatización de respuestas, chat en tiempo real, soporte multicanal y panel de métricas.
- **Mercado objetivo:** PYMEs y empresas grandes que buscan soluciones flexibles y asequibles.
- **Justificación:** Freshdesk se centra en ofrecer soluciones accesibles para PYMEs, lo que coincide con el público objetivo de SupportStation.

Zoho Desk

- **Tipo de empresa:** Privada, multinacional.
- **Sector:** Software empresarial (CRM y atención al cliente).
- **Ubicación:** India, con presencia global.
- **Tamaño:** Más de 10.000 empleados en Zoho Corporation.
- **Productos y servicios:** Gestión de tickets, automatización de tareas, chat, integraciones con CRM y herramientas de análisis.
- **Mercado objetivo:** PYMEs y empresas grandes que buscan soluciones integradas con otros servicios de Zoho.
- **Justificación:** Zoho Desk destaca por su integración con ecosistema empresarial, sirviendo como ejemplo de cómo un SaaS puede centralizar múltiples servicios en una sola plataforma.

Justificación de la empresa seleccionada

SupportStation se propone como una alternativa orientada específicamente a **PYMEs** que necesitan una solución **fácil de implementar, escalable y rentable**, enfocada en los aspectos esenciales de atención al cliente sin requerir grandes infraestructuras ni costes elevados. A diferencia de las plataformas internacionales, SupportStation se centra en:

- **Simplicidad y rapidez** de implementación.
- **Integración básica multicanal** centrada en los canales más comunes para PYMEs.
- Flexibilidad en la gestión de tickets y métricas esenciales.
- Desarrollo académico que permite experimentar e iterar rápidamente sobre nuevas funcionalidades.

De esta manera, **SupportStation** combina la **experiencia del usuario y la eficiencia operativa** con la adaptabilidad propia de un proyecto orientado a pequeñas y medianas empresas, cubriendo un nicho específico no siempre atendido por grandes plataformas SaaS.

7. Relación con los ODS

SupportStation, como plataforma **SaaS** orientada a la **atención al cliente**, se relaciona directamente con varios **Objetivos de Desarrollo Sostenible**:

ODS 8 – Trabajo decente y crecimiento económico

- Promueve la digitalización de pequeñas y medianas empresas.
- Mejora la productividad y eficiencia en la gestión de soporte.
- Facilita el crecimiento económico de negocios con recursos limitados.

ODS 9 – Industria, innovación e infraestructura

- Desarrollo e implementación de una solución tecnológica innovadora.
- Uso de tecnologías modernas y escalables (Django, React, Docker).
- Fomento de la innovación en la atención al cliente.

ODS 12 – Producción y consumo responsables

- Digitalización de procesos reduce el uso de papel y recursos físicos.

- Optimización de recursos humanos mediante automatización de tareas repetitivas.
- Promueve un uso eficiente de herramientas digitales en la empresa.

ODS 17 – Alianzas para lograr los objetivos

- Integración con APIs externas y servicios de terceros (correo, WhatsApp, redes sociales).
- Colaboración con proveedores tecnológicos y plataformas externas.

Justificación de la relación entre los ODS y la actividad de la empresa

SupportStation contribuye al cumplimiento de los **ODS** identificados porque:

- **ODS 8**: facilita el crecimiento y la competitividad de las PYMEs, permitiendo un mejor servicio al cliente con menos recursos y mayor eficiencia.
- **ODS 9**: aporta innovación tecnológica a un sector que necesita herramientas accesibles y escalables, favoreciendo la digitalización.
- **ODS 12**: reduce el consumo de papel y materiales físicos mediante la gestión digital de tickets y comunicaciones.
- **ODS 17**: fomenta alianzas y sinergias tecnológicas mediante integraciones con servicios externos que permiten mejorar la calidad del soporte.

8. Identificación de riesgos laborales

A pesar de que SupportStation es una empresa de software y no una industria con riesgos físicos intensos, la actividad principal de desarrollo y mantenimiento del software implica ciertos riesgos laborales típicos del trabajo de oficina y del sector informático. Estos riesgos se pueden clasificar en ergonómicos, psicosociales, tecnológicos y de seguridad.

1. Riesgos ergonómicos

- El trabajo frente a un ordenador durante largas jornadas puede generar:
- Posturas incorrectas (cuello, espalda y hombros).
- Dolor lumbar y cervical.
- Problemas de muñeca y manos (síndrome del túnel carpiano).
- Fatiga muscular por repetición de movimientos.

2. Riesgos derivados del uso de pantallas

- El uso prolongado de pantallas puede causar:
- Fatiga visual y sequedad ocular.
- Dolores de cabeza.
- Problemas de visión temporal.
- Sensación de cansancio y pérdida de concentración.

3. Riesgos psicosociales

- La presión por plazos, calidad del trabajo y exigencias del proyecto puede provocar:
- Estrés laboral.
- Ansiedad o agotamiento mental.
- Problemas de concentración y motivación.
- Conflictos de comunicación en el equipo.

4. Riesgos eléctricos y de infraestructura

La actividad en un entorno informático conlleva:

- Riesgo de cortocircuitos o sobrecargas por múltiples dispositivos conectados.
- Riesgo de incendio por cables en mal estado o conexiones incorrectas.
- Caídas o tropiezos por cables mal ordenados.

5. Riesgos de seguridad informática

Dado que SupportStation maneja datos de clientes y empresas, existen riesgos asociados a:

- Pérdida o fuga de datos personales.
- Accesos no autorizados o vulneraciones.
- Fallos de seguridad en el sistema o malas configuraciones.
- Uso indebido de contraseñas o credenciales.
- Conclusiones del análisis
- Síntesis de los puntos clave del análisis

El análisis de riesgos laborales para SupportStation muestra que, aunque no existen riesgos físicos graves, sí se identifican riesgos comunes del sector informático, especialmente relacionados con:

- Ergonomía y salud física, debido al trabajo prolongado frente a pantallas.
- Salud mental, por la presión de plazos y la carga de trabajo.
- Seguridad informática, por el manejo de datos sensibles y la necesidad de protección.
- Riesgos eléctricos e infraestructura, por el uso intensivo de equipos informáticos.

Para reducir y controlar los riesgos identificados, se proponen las siguientes acciones:

1. Medidas ergonómicas

- Uso de sillas y mesas ajustables.
- Altura adecuada de pantalla y teclado.
- Pausas activas cada 50–60 minutos.
- Ejercicios de estiramiento para cuello, espalda y muñecas.

2. Control del uso de pantallas

- Aplicar la regla 20-20-20 (cada 20 min mirar a 20 metros durante 20 segundos).
- Ajuste de brillo y contraste.
- Iluminación adecuada del espacio de trabajo.

3. Gestión de riesgos psicosociales

- Planificación realista del proyecto y reparto equitativo de tareas.
- Establecer horarios de trabajo y descanso.
- Comunicación clara y reuniones de seguimiento.
- Fomentar un ambiente de trabajo colaborativo.

4. Prevención de riesgos eléctricos

- Uso de regletas con protección y revisiones periódicas.
- Orden y sujeción de cables para evitar tropiezos.
- Evitar sobrecargas de enchufes.

5. Seguridad informática y protección de datos

- Aplicar políticas de contraseñas seguras y autenticación.
- Encriptación de datos sensibles.
- Copias de seguridad periódicas.
- Formación en RGPD y buenas prácticas de seguridad.

9. Tecnologías y Herramientas

Tecnologías utilizadas

El desarrollo del proyecto se apoyará en un conjunto de tecnologías modernas y ampliamente utilizadas en el ámbito profesional del desarrollo web. En el **backend** se empleará **Django**, un framework robusto y seguro que permite una rápida implementación de la lógica de negocio y la creación de APIs escalables. Para la **base de datos** se utilizará **PostgreSQL**, un sistema gestor relacional fiable y eficiente, adecuado para el manejo de grandes volúmenes de información y operaciones complejas.

En cuanto al **frontend**, se desarrollará utilizando **React**, una biblioteca de JavaScript orientada a la creación de interfaces de usuario dinámicas, reutilizables y de alto rendimiento, lo que permitirá ofrecer una experiencia de usuario fluida e intuitiva.

El sistema contará además con diversas **integraciones externas**, como una API de correo electrónico, **WhatsApp Business** y distintas **redes sociales**, con el objetivo de centralizar los canales de comunicación y mejorar la atención al cliente desde una única plataforma.

Finalmente, el **despliegue** de la aplicación se realizará mediante **Docker**, facilitando la creación de entornos consistentes, escalables y fácilmente replicables. Todas estas tecnologías han sido seleccionadas por su **robustez, escalabilidad y adecuación a los contenidos del ciclo formativo de Desarrollo de Aplicaciones Web (DAW)**, permitiendo aplicar y consolidar los conocimientos adquiridos a lo largo del ciclo.

10. Público Objetivo

El público objetivo de este proyecto está compuesto principalmente por **pequeñas y medianas empresas (PYMES)** que buscan mejorar y profesionalizar su servicio de atención al cliente sin necesidad de realizar grandes inversiones en infraestructuras tecnológicas ni en software desarrollado a medida. Este tipo de organizaciones suele contar con recursos limitados, por lo que requieren soluciones accesibles, escalables y fáciles de implementar.

El SaaS propuesto les permitirá centralizar la gestión de consultas, incidencias y comunicaciones con sus clientes, optimizando tiempos de respuesta y mejorando la calidad del servicio ofrecido. Además, al tratarse de una solución basada en la nube, las empresas podrán acceder a la plataforma desde cualquier lugar y dispositivo, adaptándose a sus necesidades de crecimiento y facilitando la mejora continua de la relación con sus clientes.

11. Modelo de Monetización

El modelo de negocio se basa en un sistema de **suscripción mensual**, con diferentes planes:

- **Plan Básico:** Funciones esenciales de tickets y chat.
 - **Plan Profesional:** Automatización y métricas avanzadas.
 - **Plan Premium:** Integraciones adicionales y soporte prioritario.
-

12. Desarrollo del proyecto (MVP y fases)

12.1 Definición del MVP

El MVP (Minimum Viable Product) de SupportStation se ha definido con el objetivo de validar la viabilidad técnica y funcional del proyecto en un periodo corto de tiempo, priorizando las **funcionalidades esenciales para una plataforma de atención al cliente**.

El MVP incluye las siguientes funcionalidades:

- **Registro y autenticación** de empresas y usuarios.
- **Gestión básica de tickets con estados** (abierto, en proceso, cerrado).
- **Chat en tiempo real entre cliente y agente**.
- **Respuestas automáticas simples para consultas frecuentes**.
- **Panel básico de métricas para seguimiento del rendimiento**.

Posibles Funcionalidades fuera del MVP (futuras)

- Integración con WhatsApp Business.
 - Integración con redes sociales.
 - Automatizaciones avanzadas.
 - Métricas detalladas (SLA, satisfacción del cliente).
 - Sistema de facturación completo.
-

12.2 Diseño de arquitectura y base de datos

La arquitectura del sistema se ha diseñado siguiendo un modelo cliente-servidor con separación clara entre frontend y backend.

Arquitectura general:

- **Frontend:** aplicación web desarrollada con React.
- **Backend:** API REST desarrollada con Django.
- **Base de datos:** PostgreSQL.
- **Comunicación en tiempo real:** Django Channels (WebSockets).
- **Despliegue:** Docker.

Entidades principales del sistema:

- **Empresa:** datos generales de la empresa cliente.
- **Usuario:** agentes y administradores asociados a una empresa.
- **Ticket:** incidencias o consultas creadas por clientes.
- **Mensaje:** mensajes intercambiados en el chat.
- **Respuesta automática:** respuestas predefinidas.
- **Métrica:** datos de rendimiento y actividad.

Este diseño permite escalabilidad, mantenimiento sencillo y futura ampliación del sistema.

Una vez definido el MVP, se procederá al diseño de la **arquitectura del sistema**, estableciendo la separación entre frontend, backend y base de datos, así como la comunicación entre los distintos componentes.

Paralelamente, se diseñará la **base de datos relacional**, definiendo las tablas principales sus relaciones y restricciones de integridad, garantizando la coherencia y seguridad de la información.

Modelo de Datos

1. Entidad Empresa

Representa a una organización cliente que utiliza la plataforma SaaS.

Constituye la entidad raíz del sistema, de la cual dependen el resto de los elementos funcionales.

Atributos principales:

- **id** (PK): Identificador único de la empresa
- **nombre**: Denominación social
- **plan**: Tipo de suscripción contratada
- **estado**: Estado operativo de la empresa dentro del sistema

Relaciones:

- Una empresa puede tener múltiples usuarios
- Una empresa puede generar múltiples tickets
- Una empresa puede registrar múltiples métricas

2. Entidad Usuario

Representa a las personas que interactúan con el sistema.

Atributos principales:

- **id** (PK): Identificador único del usuario
- **empresa_id** (FK): Empresa a la que pertenece
- **email**: Credencial de acceso
- **contraseña**: Hash de autenticación
- **rol**: Define el nivel de permisos (**admin, agente, cliente**)

Restricciones:

- Cada usuario pertenece a una única empresa
- El rol determina las acciones permitidas dentro del sistema

Relaciones:

- Un usuario puede crear tickets
 - Un usuario puede participar en mensajes
 - Un usuario puede ser asignado como agente de uno o varios tickets
-

3. Entidad Ticket

Entidad central del sistema que representa una incidencia, solicitud o problema reportado.

Atributos principales:

- `id` (PK): Identificador único del ticket
- `empresa_id` (FK): Empresa propietaria del ticket
- `cliente_id` (FK): Usuario que crea la incidencia
- `agente_id` (FK, nullable): Usuario asignado para su resolución
- `estado`: Estado del ticket (abierto, en_progreso, cerrado)
- `prioridad`: Nivel de urgencia
- `fecha_creación`
- `fecha_cierre`

Relaciones:

- Un ticket pertenece a una única empresa
 - Un ticket es creado por un cliente
 - Un ticket puede ser atendido por un agente
 - Un ticket contiene múltiples mensajes
-

4. Entidad Mensaje

Representa cada intervención dentro del canal de comunicación asociado a un ticket.

Atributos principales:

- `id` (PK): Identificador único del mensaje
- `ticket_id` (FK): Ticket al que pertenece
- `remitente_id` (FK): Usuario o sistema que envía el mensaje
- `contenido`: Texto del mensaje

- `es_automático`: Indica si el mensaje fue generado por el sistema

Relaciones:

- Un mensaje pertenece a un único ticket
 - Un ticket puede contener múltiples mensajes
-

5. Entidad RespuestaAutomática

Define reglas simples de automatización para responder mensajes sin intervención humana.

Atributos principales:

- `id` (PK): Identificador único
- `palabras_clave`: Términos que activan la respuesta
- `mensaje`: Contenido de la respuesta automática
- `activa`: Indica si la regla está habilitada

Funcionalidad:

- Cuando un mensaje entrante contiene una palabra clave configurada, el sistema genera automáticamente una respuesta asociada.
-

6. Entidad Métrica

Almacena información agregada para análisis y generación de estadísticas.

Atributos principales:

- `id` (PK): Identificador único
- `empresa_id` (FK): Empresa a la que corresponde la métrica
- `fecha`: Periodo de referencia
- `numero_tickets`: Cantidad de tickets registrados
- `tiempo_medio_respuesta`: Tiempo promedio de atención

Relaciones:

- Una empresa puede tener múltiples registros de métricas a lo largo del tiempo
-

12.3 Implementación de la versión inicial

Con la arquitectura definida, se iniciará el desarrollo de la **versión inicial del sistema**, siguiendo una metodología incremental.

Esta fase incluirá:

- Desarrollo del backend con Django.
- Configuración de la base de datos PostgreSQL.
- Implementación de la lógica de negocio.
- Desarrollo de la interfaz web.
- Despliegue del sistema mediante Docker.

La implementación se realizará de forma colaborativa utilizando **control de versiones con Git**.

12.4 Pruebas con usuarios reales y recogida de feedback

Una vez desplegada la versión inicial del sistema, se procederá a la realización de un conjunto de pruebas exhaustivas que incluirán **pruebas funcionales**, **pruebas técnicas** y **pruebas con usuarios reales**. El objetivo principal de esta fase será validar el correcto funcionamiento de la aplicación en todos sus módulos, comprobar que los requisitos definidos se cumplen adecuadamente y garantizar la estabilidad y fiabilidad del sistema. Asimismo, estas pruebas permitirán identificar posibles errores, fallos de rendimiento o problemas de usabilidad, con el fin de corregirlos y optimizar la experiencia de uso antes de su puesta en producción.

Dichas pruebas se llevarán a cabo tanto en el **entorno de desarrollo** como en un **entorno de pruebas independiente**, el cual será desplegado y gestionado mediante **Docker**, asegurando así la consistencia entre entornos y facilitando la detección temprana de incidencias relacionadas con la configuración, la infraestructura y la escalabilidad del sistema.

12.4.1 Casos de prueba funcionales

A continuación se definen algunos **casos de testing representativos** que se utilizarán para validar las funcionalidades principales del sistema:

Caso de prueba 1: Registro y autenticación de usuarios

- **Descripción:** Comprobar que un usuario puede registrarse y acceder al sistema correctamente.
- **Resultado esperado:** El usuario accede al panel principal tras introducir credenciales válidas.

- **Posibles errores detectables:** Validaciones incorrectas, errores de sesión, problemas de seguridad.

Caso de prueba 2: Creación de un ticket de soporte

- **Descripción:** Un cliente crea un nuevo ticket desde la plataforma.
- **Resultado esperado:** El ticket aparece correctamente en el panel del agente asignado.
- **Posibles errores detectables:** Fallos de guardado, datos incompletos o errores de asignación.

Caso de prueba 3: Gestión y cierre de tickets

- **Descripción:** Un agente responde a un ticket y lo marca como cerrado.
- **Resultado esperado:** El estado del ticket cambia correctamente y queda registrado en el sistema.
- **Posibles errores detectables:** Estados inconsistentes o pérdida de información.

Caso de prueba 4: Chat en tiempo real

- **Descripción:** Comunicación simultánea entre cliente y agente mediante el chat.
- **Resultado esperado:** Los mensajes se envían y reciben en tiempo real sin retrasos.
- **Posibles errores detectables:** Latencia, mensajes duplicados o pérdida de conexión.

Caso de prueba 5: Respuestas automáticas

- **Descripción:** El sistema responde automáticamente a una consulta frecuente.
- **Resultado esperado:** El cliente recibe una respuesta predefinida adecuada.
- **Posibles errores detectables:** Respuestas incorrectas o activación indebida.

Caso de prueba 6: Panel de métricas

- **Descripción:** Visualización de estadísticas de tickets.
- **Resultado esperado:** Los datos mostrados coinciden con la información real almacenada.
- **Posibles errores detectables:** Cálculos incorrectos o datos desactualizados.

12.4.2 Pruebas con usuarios reales

Además de las pruebas técnicas, se realizarán **pruebas con usuarios reales** (compañeros, familiares o usuarios simulando clientes y agentes) para evaluar la usabilidad y comprensión de la plataforma.

Los usuarios realizarán tareas como:

- Crear un ticket sin instrucciones previas.
- Utilizar el chat en tiempo real.
- Interpretar la información del panel de métricas.

12.4.3 Casos de feedback y mejora

En esta fase se **plantea la recogida de feedback** como un proceso clave para identificar posibles mejoras en la plataforma. Aunque el proyecto no se ha implementado físicamente, se ha definido cómo se evaluarían los distintos aspectos del sistema:

- **Interacción con la interfaz:** Se analizaría la facilidad para localizar funcionalidades, la claridad de los flujos de navegación y la disposición de los elementos visuales, con el objetivo de optimizar la experiencia del usuario.
- **Estados de los tickets:** Se revisará la nomenclatura y codificación visual de los estados (colores, etiquetas), asegurando que resulten claros e intuitivos para los usuarios.
- **Chat en tiempo real:** Se evaluaría la fluidez de la comunicación, identificando posibles retrasos o problemas de conexión que pudieran afectar la interacción.
- **Panel de métricas:** Se verificaría la comprensión de los datos, la claridad de los gráficos y la presentación de estadísticas relevantes, para garantizar que la información sea fácilmente interpretable.

Todo este feedback **sería documentado y analizado** para priorizar mejoras en futuras iteraciones del sistema, estableciendo un proceso de **mejora continua** que permita evolucionar la plataforma según las necesidades de los usuarios.

12.4.4 Mejora continua

El proceso de pruebas y recogida de feedback se plantea como un ciclo de **evaluación y mejora constante**. Gracias a la interacción con usuarios reales, es posible:

- Identificar errores o dificultades en la navegación.
- Mejorar la interfaz y la experiencia de usuario.
- Optimizar los flujos de tickets y la comunicación en tiempo real.

- Ajustar la presentación de métricas para facilitar su comprensión.

Este enfoque permite que el sistema evolucione progresivamente, garantizando un MVP estable, intuitivo y alineado con las necesidades reales de los usuarios.

13. Casos de Uso

Los casos de uso representan las **interacciones clave** entre los **usuarios** y el **sistema**, asegurando que el **MVP** cumpla su función principal:

- **CU1** – Registro de Empresa: Creación de cuentas para empresas.
 - **CU2** – Login de Usuario: Acceso al sistema según rol (cliente, agente, administrador).
 - **CU3** – Creación de Ticket: Apertura de incidencias por los clientes.
 - **CU4** – Asignación Automática: Distribución de tickets entre agentes.
 - **CU5** – Chat en Tiempo Real: Comunicación inmediata entre cliente y agente.
 - **CU6** – Cierre de Ticket: Finalización de la atención a un ticket.
 - **CU7** – Consulta de Métricas: Visualización de estadísticas de soporte.
-

14. Diseño de funcionalidades principales

Las funcionalidades básicas del MVP se han diseñado para cubrir las necesidades esenciales de atención al cliente:

- **Tickets:** Creación, seguimiento, asignación y cierre.
 - **Chat en tiempo real:** Comunicación directa y eficiente.
 - **Automatización:** Respuestas automáticas simples.
 - **Métricas:** Estadísticas de rendimiento y tiempos de respuesta.
 - **Integraciones futuras:** Preparación para añadir canales externos y automatizaciones avanzadas.
-

15. Tecnologías y justificación técnica

El proyecto se ha desarrollado con tecnologías **robustas, escalables y ampliamente utilizadas**:

- **Django:** Backend seguro y flexible.
 - **PostgreSQL:** Base de datos relacional fiable.
 - **React:** Interfaz moderna y reactiva.
 - **Docker:** Despliegue consistente y replicable.
 - **Django Channels:** Comunicación en tiempo real mediante WebSockets.
-

16. Seguridad y privacidad

El sistema incorpora **medidas** básicas para **proteger la información y garantizar la privacidad de los usuarios**:

- Contraseñas cifradas y control de acceso por roles.
 - Uso obligatorio de HTTPS.
 - Protección de datos personales y cumplimiento del RGPD.
 - Políticas de respaldo y recuperación de información.
-

17. Escalabilidad y mejoras futuras

SupportStation se ha diseñado con **visión de crecimiento**:

- Preparación para balanceo de **carga y microservicios**.
 - **Clasificación automática de tickets** y automatizaciones avanzadas.
 - Integraciones con WhatsApp y redes sociales.
 - Implementación futura de un sistema de facturación completo.
-

18. Modelo de negocio

El proyecto se basa en un **modelo de suscripción mensual**, adaptado a las necesidades de cada empresa:

- **Plan Básico:** Funciones esenciales (tickets y chat).
 - **Plan Profesional:** Automatizaciones y métricas avanzadas.
 - **Plan Premium:** Integraciones adicionales y soporte prioritario.
-

19. Plan de desarrollo

El desarrollo se estructuró en **cuatro semanas**, siguiendo un cronograma definido:

- **Semana 1:** Diseño de arquitectura y base de datos.
 - **Semana 2:** Desarrollo de tickets y autenticación.
 - **Semana 3:** Implementación de chat y métricas.
 - **Semana 4:** Pruebas, recogida de feedback y documentación.
-

20. Resultados y Análisis

20.1 Funcionamiento del sistema

El sistema desarrollado cumple con las **funcionalidades principales** del MVP:

- **Registro y autenticación** de usuarios por roles (empresa, agente, cliente).
- **Gestión de tickets:** creación, asignación, seguimiento y cierre sin pérdida de información.
- **Chat en tiempo real**, permitiendo comunicación fluida entre clientes y agentes.
- **Respuestas automáticas simples** para **consultas frecuentes**, reduciendo carga de trabajo.
- **Panel de métricas** que ofrece **información clara** sobre **tickets y tiempos de respuesta**, facilitando la evaluación del **rendimiento del servicio**.

20.2 Usabilidad y experiencia de usuario

La plataforma resulta **intuitiva y fácil de usar**, permitiendo a los usuarios realizar acciones básicas (crear tickets, usar el chat y consultar métricas) sin necesidad de instrucciones previas.

Se identificaron **pequeñas mejoras de interfaz**, como la claridad de los estados de los tickets y la organización visual del panel, que se podrán implementar en futuras versiones.

20.3 Rendimiento y estabilidad

- Sistema estable con múltiples usuarios simultáneos.
- Uso de Docker para entornos consistentes y fácil despliegue.
- No se detectaron fallos críticos durante las pruebas.

Durante las pruebas no se han detectado fallos críticos ni pérdidas de datos. El sistema se ha mantenido estable con varios usuarios conectados de forma simultánea, demostrando que la arquitectura definida es adecuada para un MVP.

El uso de Docker ha permitido desplegar entornos consistentes, facilitando tanto el desarrollo como las pruebas.

21. Conclusiones y Recomendaciones

21.1 Conclusiones

- **SupportStation** permite aplicar de manera práctica los conocimientos de desarrollo web aprendidos en el ciclo DAW.
- Ofrece una **solución viable y escalable** para PYMEs, centrada en **mejorar la atención al cliente**.
- La definición clara del MVP permitió **priorizar funcionalidades esenciales** y entregar un producto funcional en tiempo limitado.
- La **elección tecnológica** (Django, React, PostgreSQL, Docker) ha sido coherente con los objetivos de **robustez y escalabilidad**.
- El trabajo en equipo y uso de herramientas profesionales (Git, Trello) simula un entorno laboral real.

21.2 Recomendaciones para futuros proyectos

Como líneas de mejora y evolución futura del proyecto, se recomienda:

- Ampliar las integraciones multicanal (WhatsApp Business y redes sociales).
- Implementar métricas avanzadas y acuerdos de nivel de servicio (SLA).
- Desarrollar un sistema de facturación completo.
- Mejorar la interfaz de usuario con pruebas de usabilidad más extensas.
- Realizar pruebas de carga y rendimiento en entornos más exigentes.
- Implementar mecanismos avanzados de seguridad y auditoría.

22. Bibliografía

A continuación se incluyen las fuentes consultadas durante el desarrollo del proyecto:

1. Django Software Foundation. *Django Documentation*.
<https://docs.djangoproject.com/>
2. React. *React Official Documentation*. <https://reactjs.org/docs/getting-started.html>
3. PostgreSQL Global Development Group. *PostgreSQL Documentation*.
<https://www.postgresql.org/docs/>
4. Docker. *Docker Documentation*. <https://docs.docker.com/>
5. Zendesk. *Zendesk Product Overview*. <https://www.zendesk.com>
6. Freshworks. *Freshdesk Overview*. <https://freshdesk.com>
7. Zoho Desk. *Zoho Desk Documentation*. <https://www.zoho.com/desk/>
8. Naciones Unidas. *Objetivos de Desarrollo Sostenible (ODS)*.
<https://sdgs.un.org/goals>
9. ISO/IEC 27001: *Seguridad de la información y gestión de datos*.

23. Anexos

23.1 Documentación técnica

En este anexo se incluye la documentación técnica del proyecto, que permite comprender la estructura y funcionamiento interno del sistema:

- Diagramas de la arquitectura general (frontend, backend y base de datos).
- Esquema relacional de la base de datos.
- Descripción de las principales entidades del sistema.
- Ejemplo de configuración del entorno de despliegue con Docker.
- Capturas de pantalla de los flujos principales de la aplicación.

23.2 Ficha de autoevaluación

Nombre: Rubén Martínez Agramunt

Funciones desempeñadas:

- Planteamiento general del proyecto y definición de objetivos.
- Elaboración de la memoria técnica del proyecto.
- Organización del flujo de trabajo y gestión de tareas en Trello.

Autoevaluación:

- Se han cumplido las tareas asignadas, asegurando que el proyecto esté correctamente planificado y documentado.
- La memoria técnica refleja de forma clara y estructurada el desarrollo del proyecto, incluyendo todas las fases y apartados requeridos.

Nombre: Alejandro Tarazona Sáez

Funciones desempeñadas:

- Colaboración en el planteamiento y organización del proyecto.
- Contribución a la redacción y estructuración de la memoria técnica.
- Gestión y seguimiento de tareas en Trello, asignación de prioridades y plazos.

Autoevaluación:

- Se ha trabajado de manera eficiente en la documentación y planificación, asegurando que todos los apartados de la memoria estén completos y claros.
- La gestión de Trello permitió mantener un seguimiento ordenado del proyecto.

Aspectos a mejorar:

- Mejorar la coordinación visual de las tareas en Trello para reflejar con más claridad el progreso del proyecto.
- Profundizar en ejemplos y anexos que acompañen la memoria técnica.

23.3 Ficha de coevaluación entre miembros del equipo

Evaluador	Evaluado	Funciones	Nivel de cumplimiento	Observaciones
Rubén Martínez	Alejandro Tarazona	Planteamiento del proyecto	Excelente	Buena comunicación, colaboración constante y entrega puntual
Alejandro Tarazona	Rubén Martínez	Planteamiento del proyecto	Excelente	Trabajo responsable y eficiente en el planteamiento del sistema