



**Eloy Lozano Barrios**



**CPR Liceo La Paz**

*Desarrollo de Aplicaciones Web*

Jesús Ángel Pérez Roca

“*Technology alone is not enough – It’s technology married with liberal arts that yields us the result that makes our hears sing*.”

* Steve Jobs

# Resumen

Deskify es una plataforma web fundamentada en un sistema de ticketing destinado a la gestión de incidencias en el ámbito empresarial. El propósito de este proyecto es proveer una solución centralizada para el seguimiento y la resolución de los tickets de soporte, con el fin de optimizar la comunicación entre los usuarios finales y los agentes técnicos.

La plataforma se sustenta en tecnologías vanguardistas como React y Spring Boot, garantizando robustez, escalabilidad y una experiencia de usuario sobresaliente. Además, incorpora una arquitectura basada en microservicios, lo que permite una integración eficiente con sistemas externos y futuras ampliaciones.

Deskify emerge como resultado de la experiencia acumulada como técnico de soporte, con la intención de abordar las deficiencias identificadas en otras herramientas de helpdesk. Su diseño y funcionalidades están orientados a simplificar los procesos, reducir los tiempos de respuesta y ofrecer una experiencia más fluida y eficiente tanto para los usuarios como para los agentes técnicos. A continuación, se presentan las principales características de este software:

* Gestión de incidencias: Creación, actualización y seguimiento de los tickets.
* Roles y permisos: Diferenciación de accesos para admins, supervisores, managers, agentes y usuarios.
* Dashboard: Estadísticas en tiempo real sobre el estado de los tickets y el rendimiento del equipo.
* API Rest: Exposición de servicios para la integración con otras plataformas.
* Diseño intuitivo: Interfaz minimalista y adaptable para garantizar la accesibilidad desde cualquier dispositivo.

Algunos de los objetivos que promete cumplir Deskify son:

1. La mejora de la efectividad en la gestión de soporte técnico.
2. Reducir al máximo los tiempos de respuesta y resolución de tickets.
3. Proveer a las empresas de una herramienta que fomente la colaboración y la productividad.
4. Mejora en la satisfacción del cliente al agilizar la atención.

Por último, se detallan todas las tecnologías empleadas en el desarrollo de esta aplicación, organizadas según las diferentes partes del sistema:

* Back-End: Spring Boot, JPA, Hibernate, ModelMapper, Lombok, Rest, Swagger, Thymeleaf, JWT.
* Front-End: React.js, React Router, Axios, Material UI.
* Base de datos: MySQL.

# Abstract

**Deskify** is a web-based platform that utilises a ticketing system for the management of incidents at the enterprise level. The objective of this project is to provide a centralised solution for the tracking and resolution of support tickets, thereby enhancing communication between end users and technical agents. This platform is founded on cutting-edge technologies such as React and Spring Boot, ensuring robustness, scalability and an exceptional user experience. Additionally, it employs a microservices-based architecture, enabling efficient integration with external systems and facilitating future extensions.

The impetus for **Deskify's** development stemmed from the author's personal experience as a support technician, driven by the recognition of deficiencies in existing helpdesk tools. The design and functionalities of Deskify are meticulously crafted to streamline processes, reduce response times, and provide a more intuitive and efficient experience for both end users and technical agents. The software's primary features are as follows:

* Incident management: creation, update and tracking of tickets.
* Roles and permissions: Access differentiation for admins, supervisors, managers, agents and users.
* Dashboard: Real-time statistics on ticket status and team performance.
* API Rest: Exposure of services for integration with other platforms.
* Intuitive design: Minimalist and adaptable interface to ensure accessibility from any device.

The following objectives are proposed to be accomplished by Deskify:

1. Improving the effectiveness of technical support management.
2. The reduction of response and ticket resolution times to the maximum.
3. The provision of a tool that fosters collaboration and productivity for companies.
4. Improved customer satisfaction through the streamlining of customer service.

Finally, the technologies employed in the development of this application are listed below, organized according to the different parts of the system:

* Back-End: Spring Boot, JPA, Hibernate, ModelMapper, Lombok, Rest, Swagger, Thymeleaf, JWT.
* Front-End: React.js, React Router, Axios, Material UI.
* Database: MySQL.

# Palabras Clave

Brainstorming: herramienta de trabajo grupal que facilita el surgimiento de nuevas ideas sobre un tema o problema determinado.

HelpDesk: Servicio de soporte técnico que atiende y resuelve problemas de los usuarios finales dentro de una organización.

JavaMail: API Java que facilita el envío y recepción de correo electrónico desde código Java a través de protocolos SMTP, POP3, y IMAP.

JPA (Java Persistence API): Especificación que facilita la interacción con bases de datos relacionales mediante objetos Java.

JWT (JSON Web Tokens)**:** Tecnología utilizada para la autenticación y autorización segura en aplicaciones web, garantizando la integridad de las sesiones de usuario.

Lombok: Biblioteca de Java que simplifica el código repetitivo generando automáticamente getters, setters, constructores y otros métodos comunes.

Material UI: Biblioteca de componentes de React que permite crear interfaces modernas y visualmente atractivas.

Microservicios: Arquitectura que descompone una aplicación en servicios pequeños e independientes para mejorar su escalabilidad y mantenimiento.

ModelMapper: Herramienta que simplifica el mapeo de datos entre objetos de dominio y DTOs (Data Transfer Objects).

Must: forma básica y más directa de decir que algo es obligatorio. Significa que no tienes más remedio que hacer (o no hacer) una acción

MySQL: Sistema de gestión de bases de datos relacional que almacena toda la información de los tickets, usuarios y operaciones del sistema.

React.js: Biblioteca de JavaScript que permite construir interfaces de usuario dinámicas y altamente interactivas.

Sistema de Ticketing: Plataforma usada para gestionar incidencias, solicitudes o tareas mediante la asignación de tickets, facilitando la comunicación entre usuarios y agentes técnicos.

Spring Boot: Framework de Java para el desarrollo rápido de aplicaciones web robustas y escalables con soporte para microservicios.

# Agradecimientos

En primer lugar, es preciso expresar el más sincero agradecimiento a Mónica, mi profesora de diseño, quien ha ejercido un rol de suma importancia en el desarrollo de este proyecto desde sus inicios. Su constante apoyo y motivación, así como sus valiosas contribuciones en términos de diseño y sugerencias para la creación del mismo, han sido elementos fundamentales para la conformación de Deskify.

Asimismo, es preciso reconocer la contribución de los demás docentes, quienes, con su paciencia y dedicación, facilitaron la resolución de las dudas surgidas durante el proceso de creación. Su orientación y conocimiento resultaron fundamentales para superar los obstáculos encontrados en el desarrollo del proyecto.

Por último, es necesario expresar un agradecimiento a los compañeros Dani y Miguel, cuya ayuda, consejos y observaciones durante el desarrollo del proyecto condujeron a mejoras significativas que garantizaron el correcto funcionamiento de la plataforma. Su colaboración resultó un apoyo significativo para alcanzar este objetivo.

A todos ustedes, gracias por formar parte de este viaje.

# Contenido

[Resumen 3](#_Toc189121548)

[Abstract 4](#_Toc189121549)

[Palabras Clave 5](#_Toc189121550)

[Agradecimientos 6](#_Toc189121551)

[Contenido 7](#_Toc189121552)

[Introducción 8](#_Toc189121553)

[Motivación y objetivos 9](#_Toc189121554)

[Estado del arte 10](#_Toc189121555)

[Caso de estudio 11](#_Toc189121556)

[Implementación de tickets por correo electrónico 11](#_Toc189121557)

[Diagramas 13](#_Toc189121558)

[Diseño 14](#_Toc189121559)

[Desarrollo del Proyecto 15](#_Toc189121560)

[Manual del Administrador 16](#_Toc189121561)

[Manual del Usuario 17](#_Toc189121562)

[Viaibilidad Tecno-Económica 18](#_Toc189121563)

[Trabajo a futuro 19](#_Toc189121564)

[Conclusiones 20](#_Toc189121565)

[Biblioteca de Recursos 21](#_Toc189121566)

[Anexos 22](#_Toc189121567)

# Introducción

El desarrollo de Deskify surge como una respuesta directa a los desafíos que me he enfrentado a durante mi experiencia como técnico de soporte. Durante este período, trabajé con alguna plataforma de ticketing, enfrentándome a las limitaciones de herramientas que, aunque cumplían su función, carecían de personalización y una experiencia de usuario intuitiva. Estos mismos defectos me hicieron reflexionar sobre una posible solución, la creación de mi propio software. Una simple mejora en el diseño puede marcar la diferencia tanto para el técnico como para el usuario final.

La elección de este proyecto no fue casual, Deskify combina mi pasión por el desarrollo junto con mi conocimiento sobre el soporte técnico. Además, el desarrollo representa un desafío técnico que me permite demostrar mis conocimientos adquiridos durante este curso. Quise crear algo que no solo cumpliera con los requisitos académicos, sino también algo que me ayudase a descubrir nuevas tecnologías y formas de trabajo a la hora de la creación de un proyecto real.

En resumen, Deskify no es solo un proyecto académico; es una demostración de mi compromiso con el desarrollo y la mejora continua en el ámbito de las tecnologías de la información. Con este trabajo, aspiro a demostrar mi capacidad de aprendizaje y adaptación a los retos tecnológicos.

# Motivación y objetivos

El principal objetivo de este programa es proporcionar una solución integral que simplifique la gestión de incidencias y mejore la comunicación entre los diferentes roles involucrados en el proceso de soporte técnico. La presente plataforma es la creación de un entorno en el que la resolución de problemas sea más eficiente, lo que se traduciría en una reducción de los tiempos de respuesta y un aumento de la satisfacción de los usuarios finales.

Se identificó que las herramientas tradicionales presentan limitaciones, siendo más notable su rigidez y falta de adaptación a las necesidades específicas de cada organización. Esta constatación motivó la concepción de Deskify, fundamentada en una filosofía de personalización y escalabilidad. El presente estudio tiene como objetivo demostrar que la plataforma puede adaptarse con facilidad a diversos sectores y equipos de trabajo, ofreciendo funcionalidades modulables que permitan a las empresas configurarla según sus requisitos particulares.

En un contexto de creciente interconectividad, la accesibilidad y la usabilidad intuitiva se convierten en pilares fundamentales. En este sentido, Deskify incorpora una interfaz contemporánea y minimalista, simplificando la gestión tanto para los técnicos como para los usuarios no técnicos. Este enfoque tiene como objetivo principal reducir la curva de aprendizaje, permitiendo que cualquier individuo pueda utilizar la herramienta de manera eficiente desde el primer momento.

En suma, Deskify constituye un proyecto concebido con la finalidad de generar un impacto positivo y tangible en el ámbito del soporte técnico. El propósito de este estudio es demostrar que esta plataforma no solo constituye una solución innovadora, sino que también sirve como un ejemplo tangible de cómo la tecnología puede ser una herramienta para transformar procesos, mejorar la productividad y, en última instancia, facilitar el trabajo de las personas.

# Estado del arte

Deskify no constituye una novedad en el ámbito de las aplicaciones de ticketing, pues en la actualidad existen numerosas alternativas disponibles en el mercado, tales como *Zendesk*, *Jira Service Management*, *FreshDesk* o *Salesforce Service Cloud*. No obstante, este proyecto ha emergido con el fin de consolidarse en el mercado mediante la explotación de las debilidades de sus competidoras y la optimización de sus fortalezas, para desarrollar un software y un diseño singulares para el usuario.

A diferencia de estas herramientas, Deskify apuesta por una experiencia de usuario más accesible y personalizable, enfocándose en una interfaz minimalista y moderna, sin sacrificar la funcionalidad avanzada que exigen las organizaciones actuales. Además, incorpora flujos de trabajo intuitivos que agilizan la gestión de tickets y fomenta la colaboración eficiente entre los diferentes roles del sistema.

A continuación, se presentan las mejoras de Deskify con respecto a las alternativas existentes

* Personalización Extrema: Mientras que muchas soluciones existentes ofrecen opciones limitadas de personalización, Deskify permite a las empresas ajustar flujos de trabajo, roles, categorías y diseños según sus necesidades específicas.
* Interfaz Intuitiva: Diseñada con un enfoque moderno y minimalista, reduce la complejidad de uso, minimizando la curva de aprendizaje tanto para técnicos como para usuarios finales. Combinando lo mejor de las 4, dividiendo por páginas las diferentes funcionalidades, al contrario de *Salesforce Service Cloud*.
* Accesibilidad Multicanal Simplificada: Aunque herramientas como *Zendesk* y *Freshdesk* son multicanal, Deskify se centra en una integración más simplificada y eficiente, priorizando los canales más relevantes para cada organización.
* Roles y Permisos Flexibles: Ofrece una estructura de roles completamente adaptable, lo que facilita la asignación de responsabilidades y el seguimiento del rendimiento, algo que puede ser limitado en otras plataformas.
* Optimización de Recursos: Al estar diseñado desde cero con tecnologías modernas, - Deskify garantiza un rendimiento más eficiente y una integración fluida con herramientas externas, reduciendo tiempos de carga y mejorando la experiencia general.
* Filosofía de Mejora Continua: Deskify incorpora un enfoque ágil para adaptarse rápidamente a las nuevas tendencias y necesidades del mercado, algo que en ocasiones es más lento en plataformas consolidadas debido a su estructura compleja.

# Caso de estudio

## Implementación de tickets por correo electrónico

**1. Introducción al problema**

En el ámbito organizacional, se observa una tendencia entre los usuarios a recurrir al soporte técnico a través del correo electrónico en lugar de iniciar sesión en una plataforma específica. No obstante, múltiples herramientas de helpdesk no gestionan adecuadamente estos correos electrónicos, lo que deriva en duplicidades, pérdida de información o carencia de integración fluida. Deskify emerge como una solución a esta limitación, permitiendo que los usuarios envíen tickets directamente por correo electrónico. Estos tickets son procesados automáticamente en el sistema, garantizando la trazabilidad y el orden de los mismos.

**2. Objetivo clave**

La funcionalidad en cuestión tiene como propósito facilitar que los correos electrónicos remitidos a una dirección determinada (como, por ejemplo, soporte@deskify.com) sean procesados de manera automática y convertidos en tickets dentro del sistema de Deskify. Este proceso comprende la extracción de la información relevante del mensaje, como el asunto, el contenido y los archivos adjuntos, y su asignación a un técnico o a una cola de soporte correspondiente.

**3. Diseño de la solución**

* Recepción del correo**:** Deskify configurará un servicio que se conectará a una bandeja de entrada específica (usando protocolos como IMAP o SMTP).
* Procesamiento del contenido**:** El sistema analizará el correo para extraer:
  + **Asunto:** Será el título del ticket.
  + **Cuerpo del mensaje:** Conformará la descripción del problema.
  + **Remitente:** Para identificar o al usuario en la base de datos si está registrado.
* Creación del ticket**:** Una vez procesada la información, Deskify generará un ticket con los datos del correo, asignándolo a la categoría o técnico adecuado basado en palabras clave o reglas predefinidas.

**4. Implementación técnica**

* Conexión con el servidor de correo**:** Usar una librería como *JavaMail* para conectarse al servidor y leer los correos entrantes.
* Sincronización con la base de datos**:** Guardar el ticket y vincularlo al usuario remitente.
* Notificaciones**:** Enviar una respuesta automática al usuario confirmando que su ticket ha sido recibido.

**5. Ejemplo práctico**

Un cliente envía un correo a soporte@deskify.com con el asunto: "No puedo acceder a mi cuenta" y describe su problema en el cuerpo del mensaje.

* Deskify recibe el correo y genera un ticket con el asunto "No puedo acceder a mi cuenta".
* La descripción del problema se agrega al cuerpo del ticket.
* El cliente recibe una notificación automática con el número del ticket y un enlace para realizar el seguimiento.

**6. Resultados esperados**

* Mejora en la experiencia del usuario**:** Los clientes pueden enviar tickets sin necesidad de iniciar sesión en la plataforma.
* Mayor eficiencia**:** Reducción del tiempo de creación de tickets por parte de los técnicos.
* Centralización de datos**:** Todos los tickets generados por correo electrónico quedan integrados con el resto del sistema, facilitando su seguimiento.

# Diagramas

# Diseño

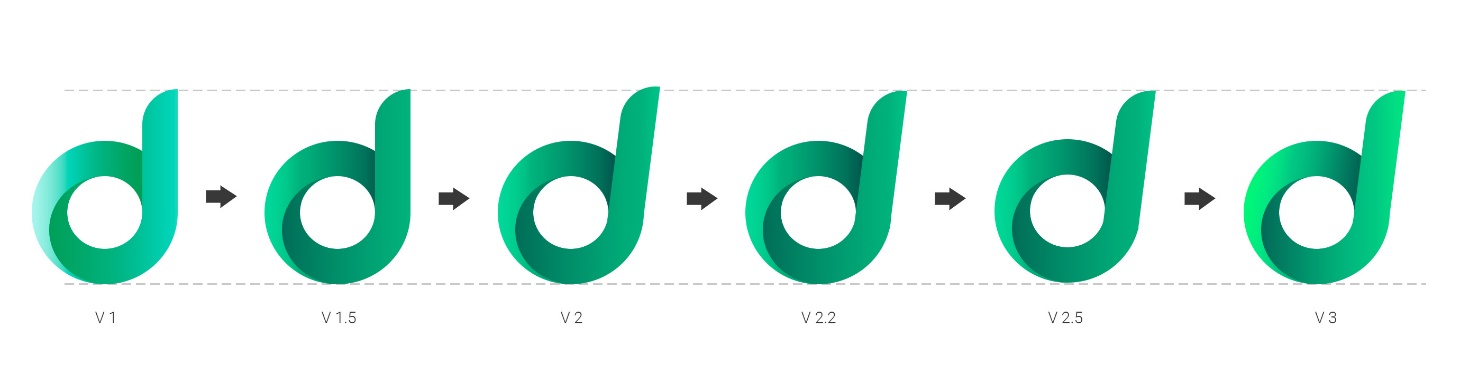
El proyecto se inició con la concepción del nombre de la marca, un elemento fundamental para definir la identidad del software. Esta tarea resultó desafiante, ya que se requería un nombre que fuera contemporáneo, accesible y cautivador, capaz de transmitir profesionalismo y facilidad de uso. Tras una exhaustiva sesión de *brainstorming*, se obtuvieron tres opciones finalistas: Deskify, EasyDesk y PriorityDesk. Se observó una tendencia hacia términos relacionados con escritorios y soporte, lo que refleja el enfoque en la gestión de tickets e incidencias.

Tras una exhaustiva deliberación, se concluyó que Deskify sobresalía como la opción más adecuada. La elección de este nombre no fue arbitraria, sino que se fundamentó en una serie de consideraciones semánticas y de usabilidad. En primer lugar, el término «desk» evoca los conceptos de soporte y organización, mientras que el sufijo moderno «-ify» sugiere transformación y dinamismo. Este nombre representa de manera precisa la visión del proyecto, que es la de transformar la experiencia del *helpdesk* en algo ágil, intuitivo y eficiente. Además, su brevedad y sonoridad lo hacen fácil de recordar y atractivo para los usuarios.

Una vez seleccionado el nombre, el próximo paso del proceso de desarrollo de la marca consistió en la creación de un logotipo que reflejara su esencia y cumpliera con los requisitos clave de simplicidad, versatilidad y adaptabilidad a colores vibrantes y tonos monocromáticos planos. Tras evaluar diversas opciones, se tomó la decisión de centrar el logotipo en la inicial de «deskify», convirtiéndola en un símbolo distintivo y memorable.

Posteriormente, se procedió a la definición de los colores que conformarían la identidad visual de la marca. Tomando en consideración el enfoque tecnológico del proyecto, se determinó que los tonos más apropiados serían el verde y el azul, ya que ambos colores suelen estar asociados con placas de circuitos y tecnología avanzada, lo que refuerza el carácter innovador de Deskify.

Durante el proceso de diseño del logotipo, mientras se examinaban bocetos y se evaluaban propuestas, emergió un elemento que se convirtió en un aspecto esencial de la composición: el logotipo debía exhibir una ligera inclinación. Este detalle no solo contribuiría a generar dinamismo, sino que también simbolizaría velocidad y eficacia, valores fundamentales de Deskify en su misión de resolver incidencias de manera rápida y efectiva.

A continuación, se presenta un gráfico que ilustra la evolución del logotipo desde sus concepciones iniciales hasta su versión actual, mostrando cómo cada iteración fue afinando la identidad visual de Deskify para alinearla con los principios mencionados.

# Desarrollo del Proyecto

La elección de las tecnologías para el desarrollo de Deskify fue clave para asegurar tanto el rendimiento como el aprendizaje durante el proceso de creación. Decidí utilizar **Spring** para el desarrollo de la **API** debido a que es la tecnología que se imparte en el curso y me brinda la oportunidad de profundizar mis conocimientos y práctica en un entorno real. Para la **landing page**, opté por **Astro**, una herramienta moderna que permite crear sitios web estáticos de alto rendimiento. Por último, para la **aplicación web** que gestionará los tickets, estoy considerando entre **Angular** y **Svelte**. Ambas son tecnologías excelentes, y mi elección final dependerá de factores como la complejidad de la interfaz y el rendimiento en tiempo de ejecución. Esta combinación de herramientas me permitirá crear una solución robusta, moderna y escalable, al mismo tiempo que continúo aprendiendo y mejorando mis habilidades de desarrollo.

**Backend - API con Spring**

La elección de **Spring** para el desarrollo de la API fue motivada por el hecho de que es la tecnología que se imparte en el curso y la cual considero ideal para reforzar mis conocimientos. Al utilizar Spring, no solo consigo implementar la funcionalidad que Deskify requiere, sino que también tengo la oportunidad de profundizar en aspectos técnicos avanzados que me permitirán mejorar mis habilidades como desarrollador. Gracias a su robustez, escalabilidad y facilidad de integración con bases de datos y otras tecnologías, Spring es la opción ideal para gestionar la lógica de negocio y la persistencia de datos en este proyecto.

**Frontend - Landing Page con Astro**

Para la creación de la **landing page** de Deskify, he decidido utilizar **Astro**, una tecnología moderna que permite la creación de sitios web rápidos y optimizados. Astro es ideal para este tipo de páginas estáticas, ya que permite integrar diversas herramientas de frontend como React, Vue o Svelte, pero en un formato altamente eficiente para el rendimiento y la carga rápida. Dado que se trata de una página estática, Astro me ayudará a mantener un desarrollo rápido y a la vez optimizar la experiencia del usuario al cargar contenido de manera eficiente.

**Frontend - Aplicación con Angular o Svelte**

En cuanto a la **aplicación web** que gestionará los tickets y las funcionalidades avanzadas de Deskify, he considerado dos opciones: **Angular** o **Svelte**. La elección de una de estas tecnologías dependerá de diversos factores durante el desarrollo:

* **Angular** es una opción robusta y madura que proporciona una gran cantidad de herramientas listas para usar, como manejo de formularios, validaciones y rutas. Es ideal para aplicaciones complejas como Deskify, ya que se adapta bien a la estructura de una aplicación escalable.
* **Svelte**, por otro lado, es un framework relativamente nuevo pero muy prometedor por su enfoque innovador. No depende de un virtual DOM y compila el código a JavaScript eficiente en tiempo de compilación, lo que mejora el rendimiento de la aplicación. Este enfoque me permitirá experimentar con una tecnología moderna que optimiza los tiempos de carga y reduce el uso de recursos.

De este modo, Deskify se desarrollará utilizando una combinación de tecnologías modernas y bien establecidas, lo que no solo garantizará un rendimiento óptimo, sino que también me permitirá adquirir y perfeccionar habilidades importantes en el desarrollo web.

# Manual del Administrador

# Manual del Usuario

# Viaibilidad Tecno-Económica

# Trabajo a futuro

# Conclusiones

# Biblioteca de Recursos

# Anexos