



TC2005B | Construcción de Software y Toma de Decisiones (Gpo 402)  
“Avance de Proyecto 4: Prueba de Concepto”

César Ignacio Saucedo Rodríguez | A01712245  
Alex Stryer Diaz | A01707173  
Aksel Deneken Maldonado | A01711966

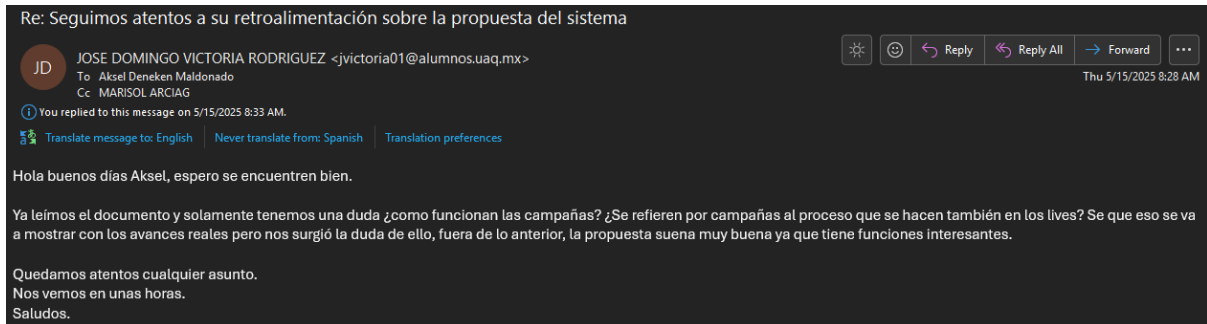
15 de Mayo de 2025

# ÍNDICE

REQUISITOS FUNCIONALES.....	3
DESCRIPCIÓN INICIAL DE LA ARQUITECTURA DE SOFTWARE.....	6
Estructura General de la Aplicación y Estilo Arquitectónico.....	6
Descomposición en Componentes Funcionales.....	7
Interfaces entre la Aplicación y los Datos Externos.....	8
¿Cómo la arquitectura soporta los requisitos no funcionales?.....	9
<b>DISEÑO Y EJECUCIÓN DE PRUEBAS.....</b>	<b>10</b>
1. Iniciar sesión en la plataforma.....	10
2. Registrar una nueva cuenta de usuario.....	11
3. Cerrar sesión.....	11
4. Editar información de perfil.....	12
5. Ver dashboard personalizado.....	13
<b>EVALUACIÓN DE USABILIDAD Y MEJORAS APLICADAS.....</b>	<b>14</b>
<b>REPORTES DEL SISTEMA.....</b>	<b>15</b>
<b>AMBIENTE DE PRODUCCIÓN.....</b>	<b>16</b>
<b>PLAN DE TRABAJO.....</b>	<b>17</b>
<b>CAMBIOS QUE SE REALIZARON EN EL PROYECTO.....</b>	<b>17</b>

## REQUISITOS FUNCIONALES

Los requisitos funcionales del sistema fueron desarrollados con el acompañamiento del socio formador, tomando en cuenta sus necesidades y los procesos clave que desean automatizar. Después de presentarlos, el socio validó cada función planteada y confirmó que la solución se alinea con lo que buscan. De hecho, nos compartieron la siguiente opinión textual: *“La propuesta suena muy buena ya que tiene funciones interesantes.”* Con base en esta retroalimentación, se definió la lista final de funcionalidades que servirán como base para el desarrollo del sistema.



ID	Nombre del Requisito	Descripción	Actor Principal	Prioridad	Dependen cias
RF01	Gestión de usuarios	Permite crear y administrar las cuentas de acceso al sistema, asignando diferentes tipos de usuario según lo que necesita hacer cada persona (por ejemplo: staff interno, clientes o el administrador de la agencia).	Administrador	Alta	-
RF02	Gestión de clientes	Permite registrar la información de las empresas que contratan los servicios de la agencia, como su nombre, datos de contacto, y responsables asignados dentro del sistema.	Staff	Alta	RF01
RF03	Gestión de campañas	Permite crear y editar campañas de marketing para cada cliente. Las campañas pueden tener tareas, archivos, responsables asignados y fechas clave.	Staff / Admin	Alta	RF02

RF04	Comentarios desde redes sociales	El sistema puede importar automáticamente los comentarios de redes sociales que tengan relación con las campañas creadas, para que el equipo los revise.	Sistema	Alta	RF03
RF05	Clasificación de comentarios	El equipo o el cliente pueden revisar los comentarios importados y clasificarlos según si son positivos, negativos o neutros, y marcar si son importantes o no.	Staff / Cliente	Media	RF04
RF06	Gestión de tareas	Permite crear tareas relacionadas con campañas, asignarlas a personas del equipo, ponerles una fecha límite y marcar su avance (como pendiente, en proceso o terminada).	Staff / Admin	Alta	RF03
RF07	Gestión de archivos	Permite subir documentos o imágenes importantes para cada campaña o cliente, y organizarlos para que se puedan consultar fácilmente.	Staff / Cliente	Alta	RF03, RF02
RF08	Filtros de información	Permite buscar y filtrar fácilmente cualquier información del sistema (como campañas, tareas, archivos, comentarios) para encontrar lo que se necesita rápido.	Todos los roles	Alta	General
RF09	Registro de actividades	Guarda automáticamente un historial con todo lo que se hace en el sistema: quién lo hizo, qué hizo y cuándo, para tener un seguimiento claro.	Sistema	Media	Todo el sistema
RF10	Inicio de sesión y recuperación	Permite que los usuarios entren al sistema de forma segura, puedan recuperar su contraseña si la olvidan y accedan solo a lo que les corresponde según su rol.	Sistema	Alta	RF01
RF11	Panel principal personalizado	Al entrar, cada tipo de usuario ve un resumen distinto con lo más importante para su rol: tareas asignadas, campañas activas, métricas, etc.	Sistema	Alta	RF01, RF03, RF06

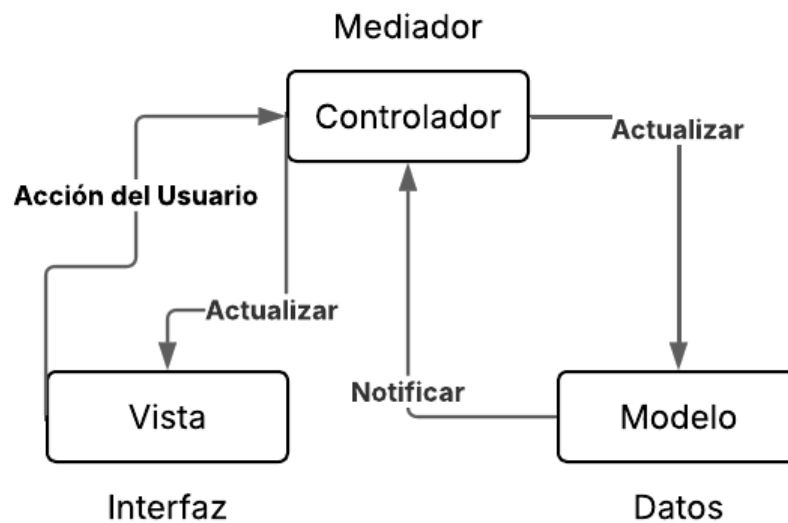
RF12	Descarga de resultados	Permite descargar en PDF o Excel la información de campañas, tareas o comentarios para guardarla, compartirla o presentarla a los clientes.	Cliente / Staff	Media	RF03, RF06, RF04
RF13	Gestión del equipo comercial	Permite registrar al personal de la agencia que se encarga de buscar o atender a los clientes, y asignarles responsabilidades.	Admin	Alta	RF01
RF14	Asignar personal a campañas	Permite vincular a miembros del equipo con campañas específicas, indicando quién es responsable de cada una.	Staff / Admin	Alta	RF13, RF03
RF15	Asignar responsable por cliente	Cada cliente debe tener asignado un miembro del equipo para que siempre se sepa quién lo atiende y se le pueda dar seguimiento.	Staff / Admin	Alta	RF02, RF13
RF16	Cambiar responsables	Si un miembro del equipo se va o cambia de puesto, se pueden reasignar fácilmente sus campañas o clientes a otra persona.	Admin	Media	RF15, RF14
RF17	Alertas del sistema	El sistema muestra avisos a los usuarios cuando tienen tareas nuevas, campañas asignadas o plazos próximos a vencer.	Todos los roles	Media	RF06
RF18	Resumen por persona	Muestra cuántos clientes o campañas tiene cada miembro del equipo, cuántas tareas ha completado y su avance general.	Admin / Staff	Alta	RF13, RF14, RF15
RF19	Historial por cliente	Muestra toda la actividad relacionada con cada cliente: campañas, archivos, tareas, comentarios o cambios importantes.	Cliente / Staff	Alta	RF02, RF04, RF06, RF07

RF20	Registro de interacciones	Permite registrar llamadas, reuniones o mensajes importantes que se hayan tenido con un cliente, para llevar un mejor seguimiento comercial.	Staff	Media	RF15
RF21	Calendario de actividades	Muestra en formato de calendario todas las tareas, reuniones o entregas, para tener una visión clara de lo que viene por día, semana o mes.	Staff / Cliente	Media	RF06, RF20
RF22	Notas internas	Permite al equipo dejar anotaciones privadas sobre un cliente o campaña, útiles para seguimiento pero que el cliente no puede ver.	Staff	Baja	RF02, RF03
RF23	Etiquetas personalizadas	Permite crear etiquetas (categorías) para clasificar campañas, clientes o tareas según cómo trabaje cada equipo.	Staff / Admin	Baja	RF02, RF03, RF06

## DESCRIPCIÓN INICIAL DE LA ARQUITECTURA DE SOFTWARE

### Estructura General de la Aplicación y Estilo Arquitectónico

El sistema que estamos desarrollando está construido con Django, un framework web en Python que principalmente sigue el patrón de diseño MVC (Modelo-Vista-Controlador).



El patrón MVC permite separar las responsabilidades del sistema en tres grandes bloques:

- **Modelo:** Se encarga de definir la estructura de los datos. Por ejemplo, qué atributos tiene un cliente, una campaña, una tarea, etc. En el proyecto esto se maneja con la base de datos. Actualizaciones, consultas, búsquedas, entre otros. Todo eso va aquí, en el modelo.
- **Vista:** Se encarga de la representación gráfica de la información. En el proyecto esto se logra a través de archivos HTML ubicados en la carpeta "templates", los cuales usan el lenguaje de plantillas de Django.
- **Controlador:** Es el cerebro del sistema, es decir, gestiona la lógica de la aplicación. Interpreta los 'requests' del usuario, accede a los datos necesarios, devuelve una respuesta y se los comunica a la vista. En Django, esta función la cumple principalmente el archivo `views.py`, y también `urls.py` y otras piezas como `apps.py` y `signals.py`.

Nuestra estructura sigue el orden que Django genera por defecto. Usamos:

- Carpeta `OBD_users/`: contiene la gestión de usuarios, sus modelos (`models.py`), lógica (`views.py`) y formularios (`forms.py`).

- Carpeta templates/: contiene la parte visual del sistema (HTML).
- Carpeta OBD\_project/: organización general de las URLs, vistas centrales, y configuración del sistema.
- Carpeta OBD/OBD/: contiene los ajustes globales como el archivo settings.py, las rutas base del sistema y los puntos de entrada (wsgi.py, asgi.py).
- Esta estructura permite que cada parte del sistema esté bien separada. Por ejemplo, si el día de mañana cambiamos la base de datos, solo tocaríamos el archivo models.py, sin necesidad de modificar el frontend o la lógica de negocio.

La arquitectura de este sistema es de tipo monolítica, lo cual significa que toda la funcionalidad está agrupada en una sola aplicación y no dividida en servicios separados.

## Descomposición en Componentes Funcionales

El sistema actual ya cuenta con varios componentes funcionales activos, y otros que están planeados para próximas etapas de implementación. Cada componente corresponde a un grupo de funcionalidades enfocadas en resolver una necesidad o acción específica dentro del sistema. En este caso se ha implementado un módulo completo que está enfocado en la gestión de usuarios.

Dentro de lo desarrollado se incluyen las siguientes funcionalidades:

- Registro de usuarios con correo electrónico, contraseña y rol asignado (cliente o vendedor).
- Autenticación de usuarios (inicio de sesión con validación).
- Verificación de correo electrónico (mediante el sistema de confirmación por email).
- Edición de perfil: el usuario puede modificar su información y su foto.
- Eliminación de cuenta: el propio usuario puede eliminar su perfil si lo desea.
- Gestión por parte del administrador: el administrador tiene la capacidad de crear, editar o eliminar usuarios desde el panel.
- Visualización personalizada según el rol: el sistema adapta la interfaz (sidebar, pestañas visibles) dependiendo de si el usuario es cliente o vendedor.



Además de lo que ya es funcional, se tiene planeada la implementación de nuevos módulos:

- Gestión de campañas: donde se crearán y administrarán campañas asociadas a clientes.
- Asignación de vendedores a campañas o clientes.
- Comentarios: permitir importar o registrar comentarios de clientes, desde redes sociales.
- Gestión de tareas: tareas asignadas a campañas, con control de estado y fecha límite.
- Dashboard con métricas y datos visuales: estadísticas para facilitar la toma de decisiones.

Estos componentes están alineados con los requisitos funcionales previamente establecidos, y su implementación será conforme vayamos avanzando en el proyecto.

## Interfaces entre la Aplicación y los Datos Externos

Actualmente, el sistema está conectado a una base de datos MariaDB, lo que permite un mejor rendimiento y escalabilidad que SQLite, la cual se estaba usando anteriormente debido a que es la que Django tiene predeterminadamente. Esta base se utiliza para almacenar toda la información relacionada con los usuarios, sus perfiles y roles dentro del sistema. En futuras etapas, también manejará los datos de campañas, tareas, comentarios, etc.

Aunque por ahora no hay conexiones activas con servicios externos, se está considerando la posibilidad de conectar el sistema con plataformas de redes sociales para importar comentarios, tener redireccionamientos a los perfiles de los clientes finales y analizar los comentarios. Esta idea todavía está en desarrollo y no se ha definido por completo, pero está contemplada dentro de los requisitos funcionales del proyecto.

Por otro lado, sí se planea integrar funciones para leer y exportar archivos Excel o CSV, principalmente para facilitar la visualización y descarga de resultados por parte del cliente. Esta funcionalidad será clave para el módulo de reportes.

En cuanto a la autenticación y validación de correos electrónicos, el sistema utiliza el paquete django-allauth, el cual gestiona el registro, login, cierre de sesión y verificación por correo electrónico de los usuarios. Más adelante se podrían integrar notificaciones o recordatorios por correo para tareas o campañas, pero esa parte aún no ha sido definida con precisión debido a su complejidad y el tiempo de implementación que conllevan.

## ¿Cómo la arquitectura soporta los requisitos no funcionales?

El sistema busca cumplir con ciertos criterios de calidad más allá de las funciones principales. Estos requisitos no funcionales cubren todo lo que se refiere a seguridad, mantenibilidad, escalabilidad y usabilidad, y han sido considerados en el enfoque técnico del desarrollo.

En cuanto a seguridad, el sistema utiliza django-allauth como base para la autenticación. Esta herramienta es, ya por defecto, bastante completa, “permite verificación de correos electrónicos, evita la enumeración de usuarios por correo, y está preparada para soportar autenticación de dos factores (2FA)” (docs.allauth.org, 2025) en caso de requerirse más adelante. Aunque todavía no se han implementado medidas avanzadas como TOTP (tokens temporales), la arquitectura permite que se agreguen en el futuro si se considera necesario.

En términos de mantenibilidad, el sistema se organiza, como se mencionó anteriormente, bajo el patrón MVC, lo que permite separar claramente la lógica de negocio (Controlador), los datos (Modelos) y la interfaz (Vista). Esto facilita que cualquier miembro del equipo pueda modificar una parte del sistema sin interferir en las demás. Mientras se mantenga y se implemente esta estructura correctamente, será sencillo escalar o modificar el sistema.

Sobre la escalabilidad, aún no se ha realizado una planeación específica relacionada con el volumen de usuarios concurrentes, ya que el sistema sigue en fase inicial y no se ha definido el proveedor de alojamiento. Sin embargo, la arquitectura permite un crecimiento progresivo sin necesidad de rediseñar todo desde cero.

En cuanto a usabilidad, la interfaz desarrollada hasta el momento es clara, ordenada y fácil de usar. Se ha trabajado con colores que destacan las acciones importantes, se ofrecen alertas llamativas, y hasta ahora los flujos de navegación son bastante intuitivos. Esto permite que cualquier usuario, sin necesidad de

capacitarse previamente, pueda entender rápidamente cómo interactuar con el sistema y cómo completar sus funciones.

## DISEÑO Y EJECUCIÓN DE PRUEBAS

Se realizaron las pruebas correspondientes a las funcionalidades implementadas hasta el momento, y todas fueron exitosas, cumpliendo con los resultados esperados en cada escenario.

### 1. Iniciar sesión en la plataforma

Escenario	Flujo Base	Alternos
Escenario 1- Inicio de sesión exitoso	Básico	-
Escenario 2- Inicio fallido por correo no registrado	Básico	A1
Escenario 3- Inicio fallido por contraseña incorrecta	Básico	A2
Escenario 4- Inicio fallido por correo no verificado	Básico	A3

Test ID	Escenario	Correo	Contraseña	Correo verificado	Resultado esperado
TC01	Login exitoso	válido	válida	válido	Accede al dashboard según su rol
TC02	Correo no registrado	inválido	N/A	N/A	Muestra error "correo no registrado"
TC03	Contraseña incorrecta	válido	inválido	válido	Muestra error "contraseña incorrecta"
TC04	Correo no verificado	válido	válida	inválido	Muestra error "verifica tu correo"

The screenshot shows the Leadly login interface. On the left is a dark sidebar with a menu: Dashboard, Usuarios, Campañas, Comentarios, Tareas, and Archivos. At the bottom of the sidebar is a link 'Cerrar sesión'. The top of the page has a dark header with the Leadly logo and links for Login and Signup. The main content area features a white login card titled 'Iniciar sesión'. It contains two input fields for 'Email address' and 'Password'. Below these fields are links for 'Forgot your password?', a 'Remember Me' checkbox, and a dark 'Iniciar sesión' button. At the bottom of the card is a link: '¿No tienes una cuenta? [Regístrate aquí](#)'.

This screenshot shows the same login page as above, but with an error message displayed in red text above the email input field: 'The email address and/or password you specified are not correct.' The email field now contains the text 'cesar2212k0934@gmail.com'. All other elements of the page, including the sidebar, header, and login form fields, remain the same.

## 2. Registrar una nueva cuenta de usuario

Escenario	Flujo Base	Alternos
Escenario 1- Registro exitoso	Básico	-
Escenario 2- Registro fallido por correo ya registrado	Básico	A1
Escenario 3-Registro fallido por campos vacíos	Básico	A2

Escenario 4-Registro fallido por rol inválido o no seleccionado	Básico	A3
---	--------	----

Test ID	Escenario	Correo	Nombre	Rol	Contraseña	Resultado esperado
TC 05	Registro exitoso	válido	válido	válido	válido	Cuenta creada, redirige a mensaje de verificación
TC 06	Correo ya registrado	Duplicado	válido	válido	válido	Muestra error "correo ya registrado"
TC 07	Campos vacíos	N/A	N/A	N/A	N/A	Muestra error "contraseña incorrecta"

Vista de registro de usuario:

Leadly Login Signup

Dashboard  
Usuarios  
Campañas  
Comentarios  
Tareas  
Archivos

Cerrar sesión

**Crear cuenta**

Email address

Username

Cliente ▾

Password

Password (again)

Registrarse

¿Ya tienes una cuenta? [Inicia sesión aquí](#)

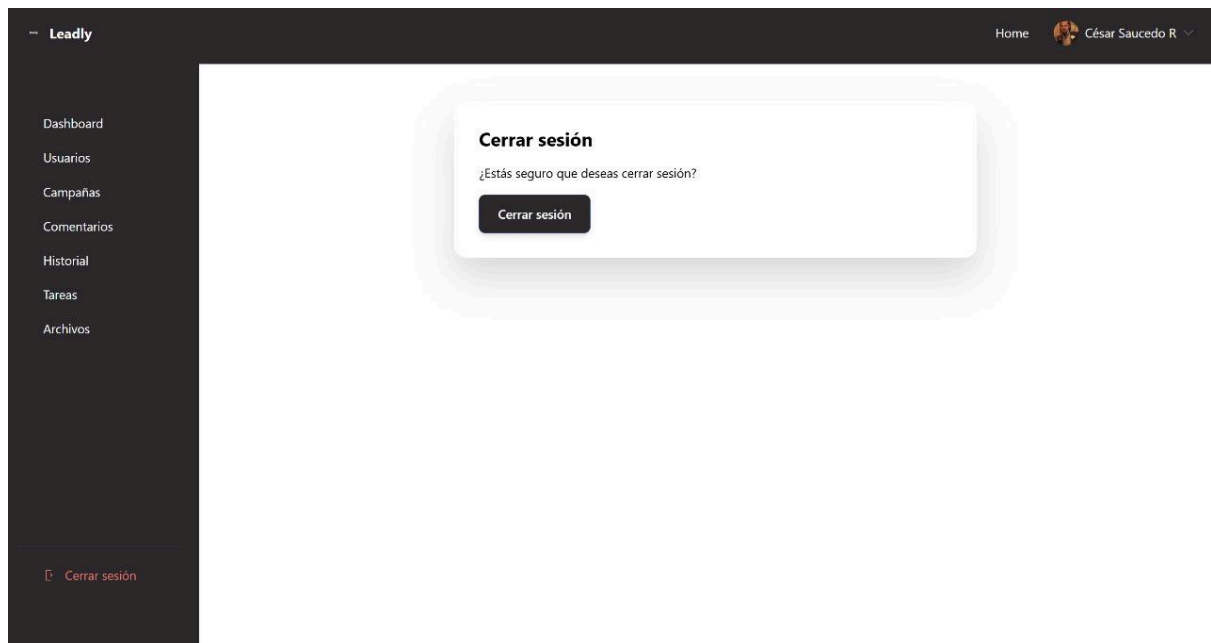
Vista de creación de perfil inválido:

### 3. Cerrar sesión

Escenario	Flujo Base	Alternos
Escenario 1-Cierre de sesión exitoso	Básico	-
Escenario 2- Intento de cierre sin sesión activa	Básico	A1

Test ID	Escenario	Sesión activa	Acción ejecutada	Resultado esperado
TC08	Cierre exitoso	Sí	Clic en cerrar	Finaliza sesión y redirige a login
TC09	Intento de cerrar sesión sin login	No	Clic en cerrar	Muestra mensaje “sesión no iniciada” o redirige

Vista de cierre de sesión:



#### 4. Editar información de perfil

Escenario	Flujo Base	Alternos
Escenario 1- Edición exitosa del perfil	Básico	-
Escenario 2- Campo de nombre vacío	Básico	A1
Escenario 3- Imagen de perfil en formato no permitido	Básico	A2
Escenario 4- No se detectan cambios para guardar	Básico	A3
Escenario 5- Error del servidor al guardar la información	Básico	A4

Test ID	Escenario	Nombre	Imagen	Cambios detectados	Resultado esperado
TC10	Edición exitosa	Válido	Válido	Sí	Perfil actualizado y mensaje de confirmación
TC11	Nombre vacío	N/A	N/A	Sí	Error “nombre no puede estar vacío”

TC1 2	Imagen no válida	Válido	Inválida	Sí	Error “formato de imagen no soportado”
TC1 3	Sin cambios detectados	Igual	Igual	No	Mensaje “no se realizaron cambios”
TC1 4	Fallo al guardar	Válido	Válida	Sí	Mensaje de error general; cambios no aplicados

Vista de edición de perfil:

Vista de edición de administrador:

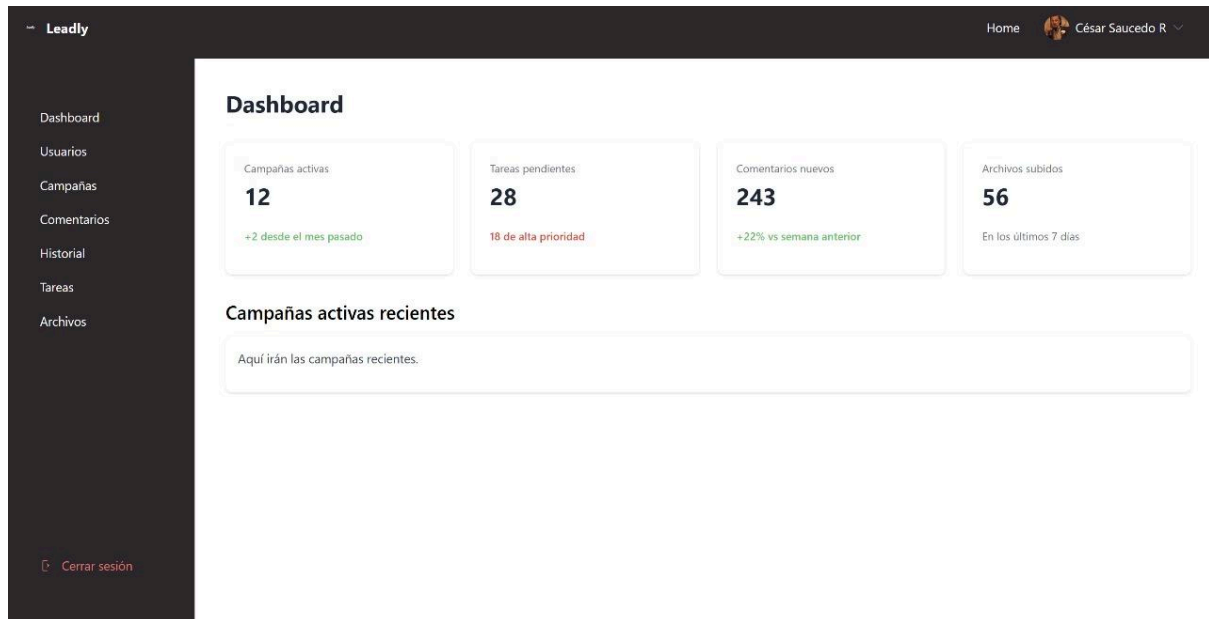


## 5. Ver dashboard personalizado

Escenario	Flujo Base	Alternos
Escenario 1- Dashboard para cliente cargado correctamente	Básico	-
Escenario 2- Dashboard para staff cargado correctamente	Básico	-
Escenario 3- Dashboard para admin cargado correctamente	Básico	-
Escenario 4- Falla al cargar el dashboard	Básico	A1
Escenario 5- Acceso forzado a otro dashboard (sin permisos)	Básico	A2

Test ID	Escenario	Rol asignado	Acceso a dashboard	Resultado esperado
TC15	Cliente accede correctamente	Cliente	Automático	Muestra dashboard con vista de cliente
TC16	Staff accede correctamente	Staff	Automático	Muestra dashboard con vista de staff
TC17	Admin accede correctamente	Admin	Automático	Muestra dashboard con vista de administrador
TC18	Rol no válido	Sin rol o rol inválido	Automático	Error o redirección a mensaje "rol no válido"
TC19	Error al cargar vista	Cualquier a	Automático	Mensaje "error al cargar dashboard"
TC20	Intento de acceder a dashboard ajeno	Cliente	URL manual	Redirección a su propio dashboard o error de acceso

Dashboard:



## EVALUACIÓN DE USABILIDAD Y MEJORAS APLICADAS

Durante el desarrollo del sistema, se presentó la interfaz a los socios formadores, quienes expresaron que les parecía moderna, clara y funcional. También se mostró a los compañeros del equipo en fase de pruebas, y la retroalimentación fue positiva en ambos casos. Debido a que no se identificaron problemas de navegación ni sugerencias importantes de cambio, no se realizaron modificaciones visuales posteriores.

A pesar de que no se aplicaron cambios, el diseño actual fue desarrollado tomando en cuenta principios clave de usabilidad. En particular, se aplicaron las siguientes reglas de oro:

- Visibilidad del estado del sistema: cada acción del usuario genera un mensaje claro en pantalla. Por ejemplo, al registrarse, editar el perfil o eliminar la cuenta, el sistema muestra confirmaciones o advertencias si hay errores o campos vacíos.
- Coherencia y estándares: se utiliza una paleta de colores armoniosa, iconografía consistente y distribución uniforme en todas las vistas. Esto ayuda a que el usuario entienda rápidamente el entorno visual.

- Control y libertad del usuario: los usuarios pueden editar su perfil, cambiar su información personal o foto, y eliminar su cuenta si lo desean. Esto les da libertad sobre su propia información.
- Prevención de errores: el sistema valida datos en formularios y avisa si hay errores antes de enviar información, como cuando una contraseña no cumple con los requisitos o se deja un campo vacío.
- Reconocimiento mejor que recuerdo: los elementos importantes están visibles y bien identificados, sin obligar al usuario a memorizar rutas o procesos.
- Flexibilidad y eficiencia: el sistema adapta la interfaz según el rol del usuario. Por ejemplo, el sidebar y el dashboard cambian según si el usuario es cliente, staff o administrador, mostrando solo lo relevante para su rol.
- Diseño estético y minimalista: se evita el exceso de contenido en pantalla. El diseño es limpio y centrado en las funciones esenciales.

Por ahora, no se ha incorporado ayuda o documentación interactiva, pero se considera como una mejora futura si el sistema se vuelve más complejo o si los usuarios lo solicitan.

## REPORTES DEL SISTEMA

El sistema contará con varios reportes que permitirán a la agencia y a sus clientes (empresas) tener un panorama claro de su desempeño, tareas pendientes, métricas internas y conversiones logradas. Estos reportes están pensados para facilitar la toma de decisiones tanto a nivel operativo como comercial.

Los reportes estarán disponibles para los distintos tipos de usuarios con los permisos adecuados, y se podrán descargar en formato Excel (para análisis más detallado mediante fórmulas) o PDF (para compartirlos de forma formal).

Nombre del Reporte	¿Quién lo puede ver?	Campos principales	Visualización
--------------------	----------------------	--------------------	---------------

Reporte de tareas por campaña	Staff / Cliente	Nombre de campaña, tarea, estado (pendiente/en proceso/terminada), fecha de entrega, responsable	Tabla
Reporte de usuarios interesados	Admin / Empresa cliente	Nombre, correo, red social de origen, comentario, fecha, campaña asociada	Tabla
Reporte de conversiones	Admin / Empresa cliente	Número total de comentarios, número de leads, número de ventas, tasa de conversión (%)	Tabla + gráfico de barras
Reporte de ventas totales por campaña	Admin / Cliente	Campaña, vendedor asignado, total ventas, total ingresos	Tabla
Reporte de rendimiento de vendedores	Admin / Staff	Nombre del vendedor, campañas asignadas, ventas logradas, tareas pendientes	Tabla + gráfico de barras
Reporte de comentarios clasificados	Staff / Cliente	Comentario, campaña, sentimiento (positivo/negativo/neutro), clasificado por	Tabla
Reporte de historial por cliente	Admin / Staff / Cliente	Cliente, campañas asociadas, tareas cumplidas, interacciones, archivos subidos	Tabla

## AMBIENTE DE PRODUCCIÓN

Para el despliegue del sistema en un entorno real, se considera utilizar como proveedor principal Amazon Web Services (AWS), ya que es una opción confiable, escalable y compatible con las tecnologías utilizadas en el proyecto (Django + MariaDB).

Aunque todavía no se ha concretado el plan de hosting específico, AWS ofrece opciones flexibles que permiten comenzar con un nivel gratuito o de bajo costo y escalar según sea necesario. Esta opción también garantiza disponibilidad, soporte técnico y posibilidad de integrar servicios adicionales como almacenamiento, backups automáticos y análisis.

En cuanto al dominio, se propondrá al socio formador que el sistema se publique bajo un nombre personalizado con relación a su agencia. Esto le daría mayor profesionalismo y facilitaría el acceso de sus clientes. Sin embargo, en fases

iniciales del proyecto se puede trabajar con un subdominio proporcionado por el proveedor.

Dado que el sistema maneja información personal (usuarios, correos, datos de campañas), es obligatorio utilizar el protocolo HTTPS para proteger todas las comunicaciones. Se contempla integrar un certificado SSL desde el inicio del despliegue. En caso de que AWS no lo proporcione de forma gratuita, se evaluará utilizar Let's Encrypt, una solución ampliamente aceptada y sin costo que permite emitir certificados válidos para HTTPS.

Esta decisión se comentará con el socio formador para definir juntos el alcance y la inversión que implicaría mantener el sistema en línea de forma segura y estable.

## PLAN DE TRABAJO

Actividad	Responsable	Tiempo Estimado (hrs)	Tiempo Real (hrs)	¿Qué se logró?
Replanteamiento del proyecto	Alex	.5	.5	Ajuste de objetivos, alcances y enfoque técnico del sistema.
Actualización del documento de avance 2	Alex	2	3	Creación de nuevos diagramas, requisitos, reglas de negocios, etc
Creación del modelo MER	Alex	.5	.5	Rediseño estructural con base en nueva lógica de usuarios.
Creación de nuevas interfaces	Alex	1.5	2	Diseño visual de nuevas pantallas (login, dashboard, etc.).
Sistema de autenticación (backend)	César	2	2	Registro, login, verificación con django-allauth.
Creación de estilos (CSS)	César	2.5	3	Adaptación responsive y estética general.
Creación de la base de datos (MariaDB)	César	2	2	Migraciones, creación de tablas, pruebas de conexión.
Avance General del Sitio	César/Alex/Aksel	5	7	Creación de una página con diseño limpio y moderno/completo
Contacto con socio formador	Aksel	0.5	0.5	Se compartieron los casos de uso.

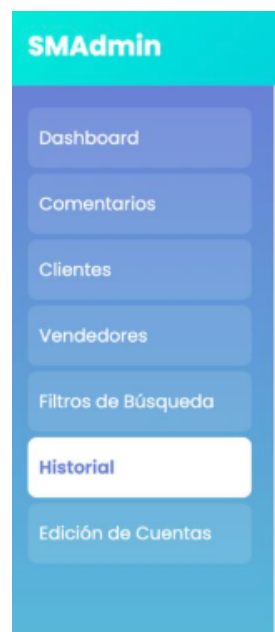
Casos de uso y diagramas	Aksel	1	1.5	Diagramas claros para funcionalidad principal.
Documento de avance 4	Aksel/Alex	2.5	3	Redacción técnica, arquitectura, requisitos.
Presentación del avance	Aksel	1	1	Diapositivas claras, incluye todos los elementos clave.

## CAMBIOS QUE SE REALIZARON EN EL PROYECTO

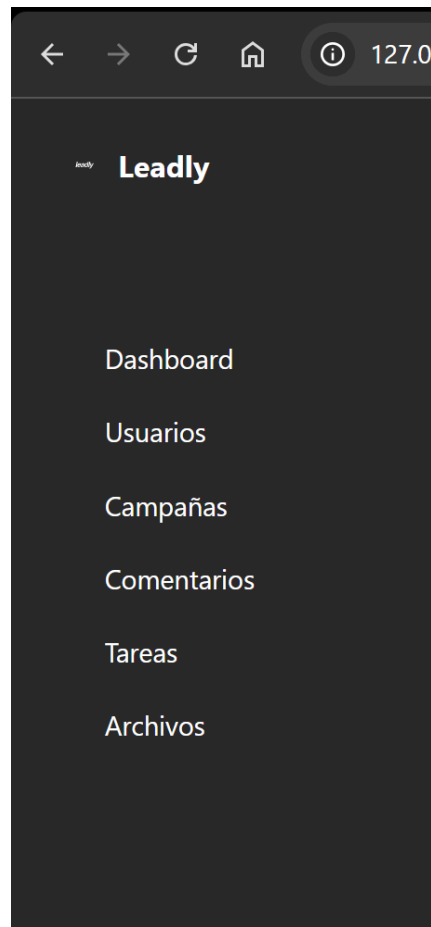
### 1. Interfaces diferenciadas por rol

- **Antes:** Se había considerado una vista general única para todos los usuarios, con contenido filtrado según su rol.
- **Mejora aplicada:** Se implementaron vistas completamente diferenciadas para *administrador, cliente y staff*, aprovechando el panel administrativo de Django para gestión avanzada y adaptando la navegación y contenido a cada tipo de usuario.
- **Justificación (Regla aplicada):** *Visibilidad del estado del sistema / Reconocimiento en lugar de recuerdo.* Los usuarios solo ven lo relevante para su rol, reduciendo sobrecarga cognitiva.

### ANTES:



## DESPUÉS:



### 2. Autenticación robusta con django-allauth

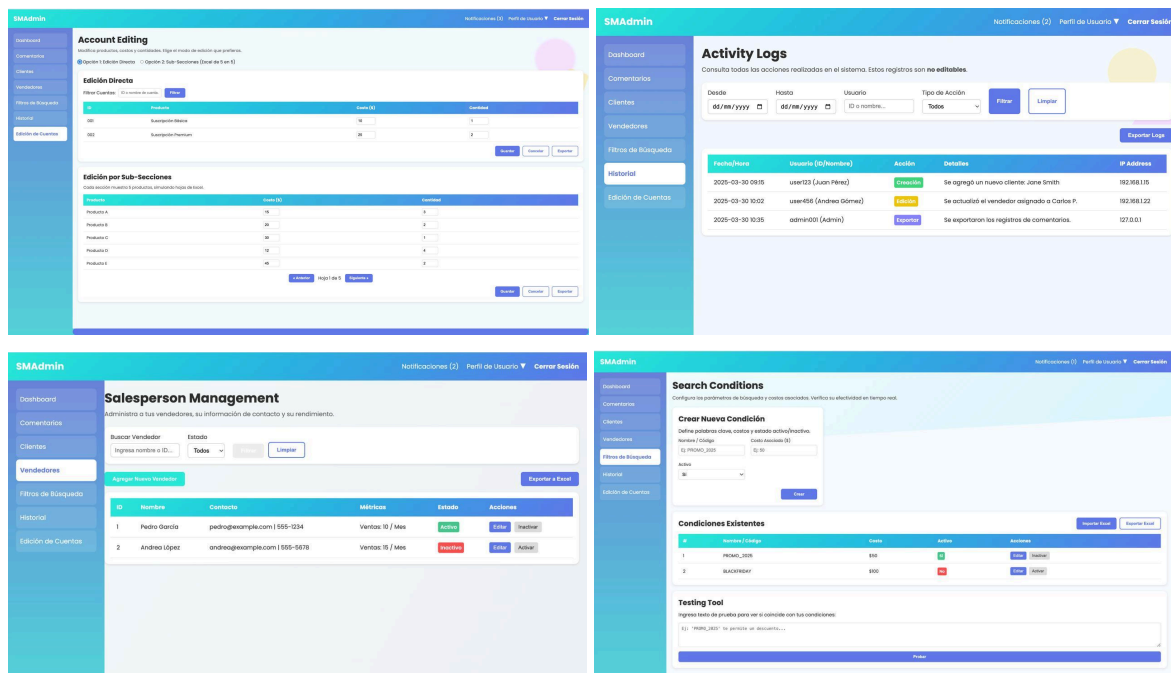
- **Antes:** Se iba a usar un sistema de autenticación básico manual.
- **Mejora aplicada:** Se integró django-allauth, un sistema avanzado que incluye login, registro, recuperación y verificación por correo electrónico de manera segura y eficiente.
- **Justificación (Regla aplicada):** *Prevención de errores / Control y libertad del usuario.* Se garantiza seguridad, claridad en flujos y retroalimentación inmediata en el proceso de autenticación.

### 3. Rediseño visual inspirado en marcas modernas

- **Antes:** El diseño inicial era funcional, pero visualmente genérico.

- **Mejora aplicada:** Se rediseñó la interfaz utilizando una paleta moderna, fuentes limpias y referencias visuales tomadas de marcas como **Nike, Puma, Under Armour y Apple**, buscando una estética **elegante, limpia y profesional**.
- **Justificación (Regla aplicada):** *Diseño estético y minimalista*. La nueva interfaz mejora la percepción del producto, genera confianza y facilita el enfoque en lo importante.

Antes:






Despues:

Leadly

Dashboard
Usuarios
Campañas
Comentarios
Tareas
Archivos

Cerrar sesión

Home  Lead

### Dashboard

Campañas activas

12

+2 desde el mes pasado

Tareas pendientes

28

18 de alta prioridad

Comentarios nuevos

243

+22% vs semana anterior

Archivos subidos

56

En los últimos 7 días


### Campañas activas recientes

Aquí irán las campañas recientes.


Leadly

Dashboard
Usuarios
Campañas
Comentarios
Tareas
Archivos

Cerrar sesión

Home  Lead

### Edit your Profile



**Lead**  
@leadlyadmin

Seleccionar archivo | Ningún archivo seleccionado

Lead

Anadir Informacion

Submit

Cancel