

# TRABAJO FIN DE GRADO



**ALONSO  
DE AVELLANEDA**

**Alonso de Avellaneda**

**Técnico Superior en Desarrollo de  
Aplicaciones Multiplataforma**

**XCRM.net**

**Autores:**

Espartaco Nassekine Maksimov  
Antonio Águila Alcaraz  
Daniel Téllez Corona

**Director:** Juan José Rodríguez Castilla

**Tutor:** Alejandro de Acuña Jiménez

Alcalá de Henares, junio de 2025

# XCRM - Sistema CRM Multi-Tenant

## Documentación Técnica Completa

### ÍNDICE

1. Resumen Ejecutivo
2. Introducción
3. Análisis del Entorno
4. Solución Propuesta
5. Tecnologías Utilizadas
6. Planificación Temporal
7. Diseño e Implementación
8. Manual del Usuario
9. Conclusiones
10. Bibliografía

## 1. RESUMEN EJECUTIVO

Español

El proyecto XCRM es una aplicación CRM (Customer Relationship Management) desarrollada con Spring Boot que permite gestionar campañas, clientes, interacciones y generar métricas y reportes. Diseñada para organizaciones que requieren una solución multi-tenant, genera una base de datos por organización al registrarse automáticamente. La aplicación utiliza tecnologías modernas como Spring Security para la autenticación, Thymeleaf para las vistas, Bootstrap para el diseño responsivo y JasperReports para la generación de reportes. El objetivo principal es proporcionar una herramienta eficiente para la gestión de relaciones con los clientes, mejorando la productividad y la toma de decisiones.

English

The XCRM project is a CRM (Customer Relationship Management) application developed with Spring Boot that allows managing campaigns, customers, interactions, and generating metrics and reports. Designed for organizations requiring a multi-tenant solution, it automatically generates a database per organization upon registration. The application uses modern technologies such as Spring Security for authentication, Thymeleaf for views, Bootstrap for responsive design, and JasperReports for report generation. The main objective is to provide an efficient tool for managing customer relationships, improving productivity and decision-making.

## 2. INTRODUCCIÓN

### 2.1 Contexto del Proyecto

En un entorno empresarial cada vez más competitivo, la gestión eficiente de las relaciones con los clientes se ha convertido en un factor clave para el éxito. Las empresas necesitan herramientas que les permitan organizar y analizar las interacciones con sus clientes de manera efectiva. El proyecto XCRM surge como respuesta a esta necesidad, proporcionando una solución CRM escalable y personalizable.

### 2.2 Justificación y Relevancia

El desarrollo de XCRM se justifica por la creciente demanda de soluciones CRM que sean:

- Flexibles: Adaptables a diferentes tipos de organizaciones
- Seguras: Con sistemas robustos de autenticación y autorización

- Escalables: Capaces de crecer con las necesidades del negocio
- Multi-tenant: Optimizando recursos y reduciendo costos operativos

### 2.3 Objetivos Generales y Específicos

Objetivo General: Desarrollar una aplicación CRM multi-tenant que facilite la gestión de campañas, clientes e interacciones, proporcionando herramientas de análisis y reporting para mejorar la toma de decisiones empresariales.

Objetivos Específicos:

- .1 Implementar autenticación y autorización seguras con Spring Security
- .2 Diseñar una interfaz intuitiva y responsiva utilizando Thymeleaf y Bootstrap
- .3 Desarrollar un sistema multi-tenant con bases de datos independientes por organización
- .4 Generar reportes automatizados en múltiples formatos (PDF, Excel, HTML)
- .5 Garantizar la escalabilidad del sistema mediante una arquitectura modular
- .6 Implementar métricas y dashboards para análisis de rendimiento
- .7 Asegurar la calidad del código mediante pruebas unitarias

### 2.4 Requisitos del Sistema

Requisitos de Hardware:

- Servidor con mínimo 1 GB de RAM (recomendado 4 GB)
- Procesador con al menos 1 núcleos
- Espacio en disco: 500mb mínimo
- Conexión a internet estable

Requisitos de Software:

- Java 17 o superior
- MySQL 8.0 o MariaDB 10.6+
- Git para control de versiones
- Maven 3.6+ (incluido en el wrapper del proyecto)

### 2.5 Restricciones y Limitaciones

- Requiere conocimientos técnicos para la configuración inicial
- Limitado a bases de datos MySQL/MariaDB
- El sistema multi-tenant requiere permisos de creación de bases de datos
- Las funcionalidades de reporting requieren JasperReports configurado correctamente

## 3. ANÁLISIS DEL ENTORNO

### 3.1 Estudio de Viabilidad

Viabilidad Técnica: El proyecto es técnicamente viable utilizando tecnologías maduras y estables como Spring Boot, MySQL y JasperReports. La arquitectura multi-tenant ha sido validada en múltiples proyectos empresariales.

Viabilidad Económica: El uso de tecnologías de código abierto reduce significativamente los costos de licenciamiento. El modelo multi-tenant optimiza el uso de recursos de hardware.

Viabilidad Operacional: La interfaz intuitiva y la documentación completa facilitan la adopción por parte de los usuarios finales.

### 3.2 Análisis de Requisitos

## Requisitos Funcionales

Código	Descripción
RF01	Conseguir un sistema multitenant basado en independencia de DB
RF02	Sistema de autenticación y autorización por roles, bloqueos de IP
RF03	Gestión completa de campañas comerciales
RF04	Registro y seguimiento de clientes
RF05	Gestión de interacciones cliente-comercial-empresa
RF06	Generación de reportes en múltiples formatos
RF07	Dashboard con métricas clave
RF08	Asignación de clientes a campañas
RF09	Seguimiento de resultados por campaña, comercial
RF09	Conseguir un sistema de notificación interno y para clientes

## Requisitos No Funcionales

Código	Categoría	Descripción
RNF01	Seguridad	Encriptación de contraseñas (bcrypt) y autenticación JWT
RNF02	Rendimiento	Tiempo de respuesta de endpoints < 2 segundos
RNF03	API Design	Endpoints RESTful bien estructurados y documentados
RNF04	Escalabilidad	Soporte para múltiples organizaciones con tenant isolation
RNF05	Disponibilidad	99% de tiempo activo considerando límites tier gratuito
RNF06	Mantenibilidad	Controllers organizados, separación de responsabilidades
RNF07	Base de Datos	Consultas optimizadas para RDS gratuita (20GB), índices apropiados
RNF08	Logging	Logs estructurados para debugging y monitoreo básico
RNF09	Validación	Validación de datos de entrada en todos los endpoints
RNF10	Charts Data	Endpoints específicos optimizados para Chart.js con datos agregados
RNF11	Rate Limiting	Límites de requests por IP/usuario para proteger recursos gratuitos
RNF12	Error Handling	Manejo consistente de errores con códigos HTTP apropiados
RNF13	Caching	Cache básico para consultas frecuentes (datos de charts), reportes
RNF14	Memory Usage	Optimización para 1GB RAM de EC2 t2.micro
RNF15	Backup Strategy	Estrategia de respaldo compatible con RDS gratuita

### 3.3 Cronograma del Proyecto:

Fase	Fecha	Días restantes	Estado
Análisis, Prototipo UI...	24/03/2025	78 días	✅ Completado
Módulo Core Multi-tenant	18/04/2025	53 días	✅ Completado
Dashdoard funcional	20/05/2025	21 días	✅ Completado
Sistema de Reportes	25/05/2025	15 días	✅ Completado
Pruebas y Depuración	28/05/2025	12 días	✅ Completado
Documentación Final	09/06/2025	1 días	🕒 Pendiente
Defensa del Proyecto	10/06/2025	0 día	🎯 Objetivo

### 3.4 Actores del Sistema

Actor Primario: Administrador de Organización

- Registro de nueva organización
- Configuración del sistema
- Gestión de usuarios

- Acceso completo a reportes y métricas

Actor Secundario: Usuario Comercial

- Gestión de campañas asignadas
- Registro de interacciones con clientes
- Consulta de métricas básicas
- Generación de reportes de sus campañas

### 3.5 Análisis de Riesgos

Riesgo	Probabilidad	Impacto	Estrategia de Mitigación
Incompatibilidad entre JasperReports y Spring Boot	Baja	Medio	Pruebas tempranas de integración, prueba de dependencias, pruebas en el servidor.
Sobrecarga de base de datos con múltiples tenants	Media	Alto	Optimización de consultas y índices, uso de AWS RDS, escalar si es necesario
Problemas de seguridad en autenticación	Baja	Muy Alto	Implementación de Spring Security con mejores prácticas, bloqueo por IP

## 4. SOLUCIÓN PROPUESTA

### 4.1 Arquitectura del Sistema

#### 1. Capa de Presentación:

- ☒ API REST con versionamiento (/api/v1)
- ☒ Controladores con manejo de excepciones
- ☒ DTOs para transferencia de datos (records Java 17+)
- ☒ Validación con Bean Validation 3.0 (@Valid)

#### Capa de Negocio:

- ☒ Interfaces/Implementaciones (Principio DIP)
- ☒ Servicios transaccionales (@Transactional)
- ☒ Comunicación asíncrona (Eventos/Spring Integration)

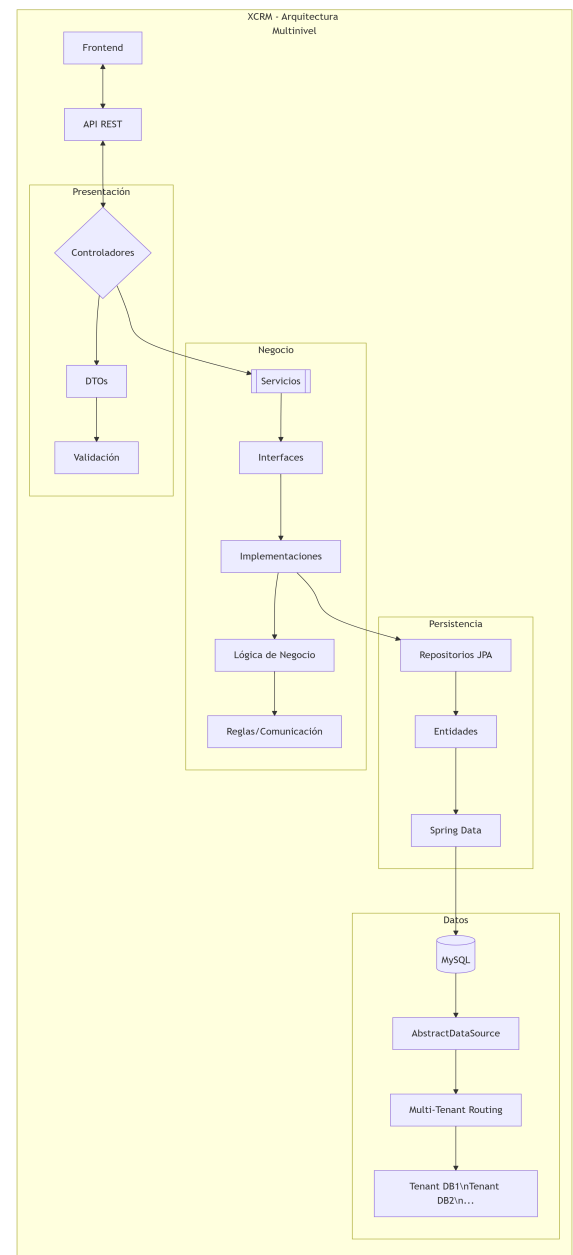
#### Capa de Persistencia:

- ☒ Repositorios JPA (Spring Data + Hibernate)
- ☒ Entidades con mapeamiento avanzado (@ManyToOne, @JoinTable)

#### Capa de Datos:

- ☒ QueryDSL para consultas dinámicas
- ☒ AbstractDataSource con enrutamiento multi-tenant
- ☒ Connection Pooling (HikariCP)
- ☒ Esquema por Tenant (Strategy Pattern)

Patrones aplicados:



- SOLID (Single Responsibility en servicios)
- Strategy (Multi-Tenancy)
- Factory (DataSource dinámico)

## 4.2 Modelo Multi-Tenant

Estrategia: Database per Tenant

Cada organización que se registra en el sistema obtiene automáticamente:

- Base de datos dedicada con nombre único
- Esquema completo de tablas
- Configuración de conexión independiente
- Aislamiento completo de datos

Ventajas:

- Máximo aislamiento de datos
- Facilidad para backups individuales
- Escalabilidad independiente por tenant
- Cumplimiento de normativas de privacidad

## 4.3 Estructura de Bases de Datos

 Base de Datos Principal (mi\_app)

Contiene información global sobre las organizaciones registradas y sus respectivas configuraciones de conexión.

Tabla: organizaciones

- id (PK): Identificador único de la organización.
- nombre: Nombre de la organización.
- nombreDB: Nombre de la base de datos asociada.
- email: Correo de contacto.
- plan: Tipo de suscripción.
- creado: Fecha de creación.

Tabla: tenant\_configs

- organization\_id (FK): Clave foránea hacia organizaciones.
- db\_host: Host de conexión a la base de datos.
- db\_port: Puerto de conexión.
- db\_username: Usuario.
- encrypted\_password: Contraseña cifrada.

 Base de Datos por Tenant

Cada organización tiene su propia base de datos, con tablas específicas para la gestión CRM.


 Seguridad y Autenticación

Tabla: users

- id (PK): UUID del usuario.
- username, password, enabled
- organizacion\_id (FK)
- fotoUrl: Foto de perfil (opcional)

Tabla: authorities

- id (PK)
- user\_id (FK hacia users)
- authority: Rol asignado al usuario



## Gestión de Campañas y Clientes

Tabla: campanias

- id (PK), nombre, descripcion
- fecha\_inicio, fecha\_fin
- organizacion\_id (FK)

Tabla: clientes

- id (PK), nombre, email, telefono
- direccion, organizacion\_id (FK)

Tabla: clientes\_campanias

- campania\_id, cliente\_id (PK compuesta)
- Relaciones N:M entre campañas y clientes

Tabla: comerciales\_campanias

- campania\_id, comercial\_id (PK compuesta)
- Asignación de comerciales a campañas

Tabla: comerciales\_clientes

- cliente\_id, comercial\_id (PK compuesta)
- Relación entre comerciales y sus clientes



## Interacciones y Ventas

Tabla: interacciones

- id (PK), fecha\_hora, estado, tipo, notas
- campania\_id, cliente\_id, comercial\_id (FK)

Tabla: ventas

- id (PK), fecha, producto, monto
- interaccion\_id (FK, relación 1:1)

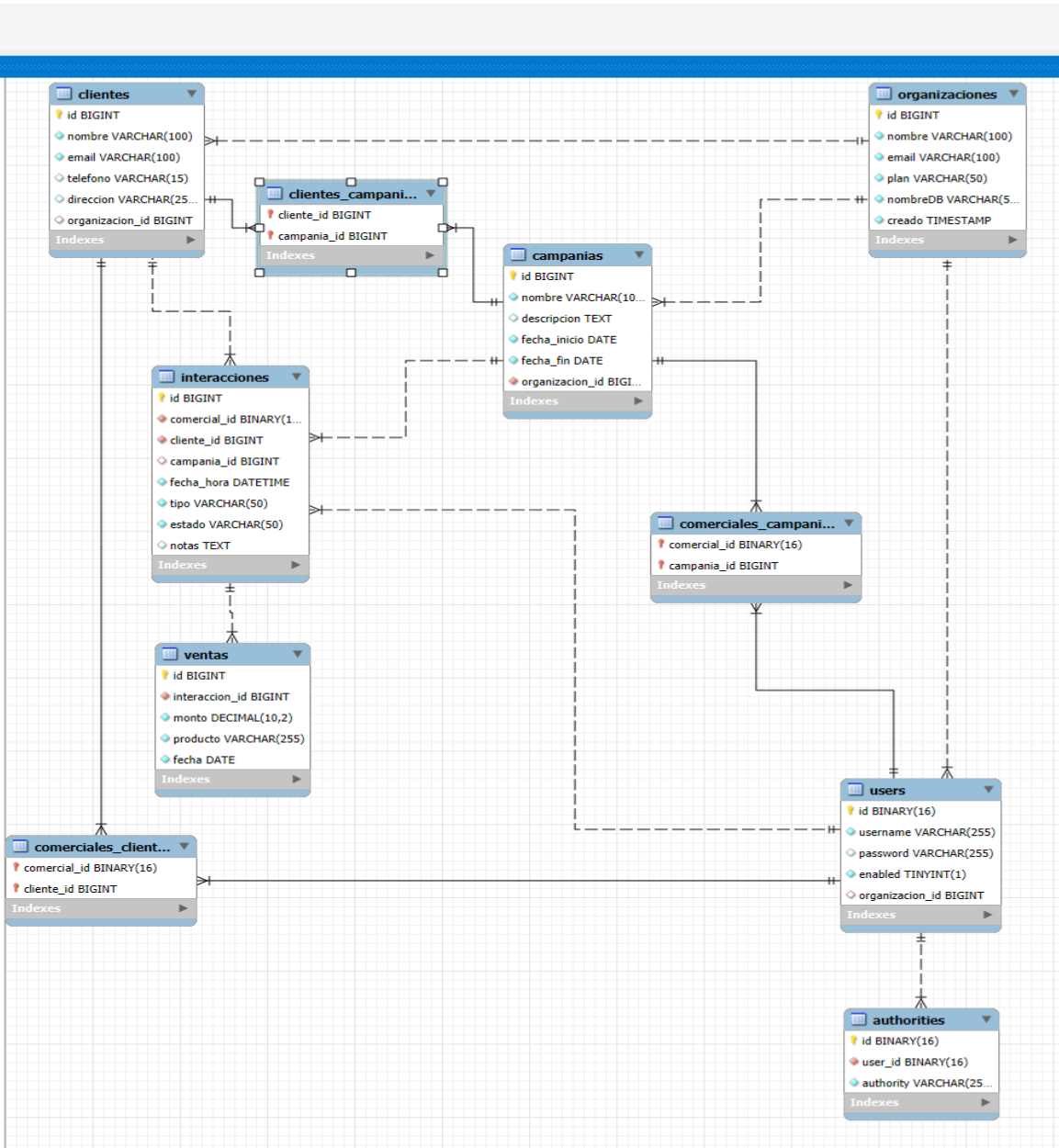


## Contacto

Tabla: contacto\_mensaje

- id (PK), nombre, email, asunto, mensaje

- `archivoNombre`, `archivoDropboxUrl`
- `fechaEnvio`, `usuarioId`



#### 4.4 Flujo de Trabajo

## 1. Registro e Inicialización de la Organización

- Usuario → Formulario de Registro
- Se recogen los datos básicos de la organización y del usuario administrador.
- Validación de datos



- Creación de la Organización y su Base de Datos
- Se genera automáticamente una base de datos exclusiva para la organización.
- Creación del Usuario Administrador
- Este usuario tiene acceso total para gestionar el entorno de su organización.
- Configuración Inicial del Sistema
- Se cargan configuraciones base y se habilita el acceso a los módulos.

## **2. Gestión Administrativa y Operativa**

Accesible tras el login del usuario administrador:

- Usuarios
- Creación y gestión de cuentas internas (comerciales, analistas...).
- Asignación de roles y permisos.
- Clientes
- Registro y mantenimiento de información comercial de los clientes.
- Posibilidad de asignar clientes a campañas y comerciales.
- Campañas
- Creación de campañas de marketing/ventas.
- Asignación de campañas a usuarios comerciales.
- Asociación opcional de clientes a campañas específicas.

## **3. Interacciones Comerciales**

- Selección de Campaña y Cliente
- El comercial accede a las campañas asignadas.
- Puede elegir un cliente para interactuar.
- Registro de Interacción
- Modalidad: llamada, email, mensaje, visita presencial.
- Estado: exitosa, en proceso, no contactado, etc.
- Detalles: observaciones, resultados y comentarios.
- Historial de Interacciones
- Visualización del histórico completo de interacciones por cliente.
- Incluye todas las campañas y comerciales involucrados.

## **4. Ventas**

- Acceso desde la sección de Ventas
- Selección de campaña y cliente.
- Vinculación directa con una interacción previa.
- Registro del producto vendido y el monto.
- Análisis de Rendimiento

- Cada venta queda asociada a una interacción y alimenta los reportes.

## 5. Dashboard y Métricas

- Visualización de KPI
- Uso de gráficos generados con Chart.js y Jasper Report
- Métricas de rendimiento: interacciones, campañas activas, tasas de conversión, etc.
- Reportes Exportables
- Descarga de informes PDF generados con JasperReports.
- Reportes disponibles:
  - Interacciones por comercial.
  - Todos los clientes.
  - Todos los usuarios de la organización.

## 4.5 Seguridad Implementada

La capa de seguridad del sistema ha sido diseñada siguiendo buenas prácticas de desarrollo seguro, garantizando la integridad, confidencialidad y disponibilidad de los recursos y datos sensibles. A continuación se detalla cada uno de los componentes implementados:

Autenticación:

- Formularios de autenticación personalizados

Se utiliza Spring Security para gestionar la autenticación mediante formularios web configurados específicamente para las necesidades del proyecto.

- Encriptación de contraseñas con BCryptPasswordEncoder

Todas las contraseñas son almacenadas utilizando un algoritmo de encriptación robusto y seguro basado en BCrypt, evitando riesgos asociados al almacenamiento en texto plano o en formatos débiles.

- Gestión segura de sesiones

Las sesiones son manejadas de forma segura, incluyendo la invalidación automática tras cerrar sesión y la protección contra fijación de sesión (session fixation protection).

- Logout automático por inactividad

El sistema invalida automáticamente las sesiones de usuario tras un periodo de inactividad prolongado, mejorando la seguridad en entornos compartidos.

## Autorización:

### - Control de acceso basado en roles (RBAC)

El acceso a funcionalidades críticas del sistema está restringido según el rol del usuario (por ejemplo: ADMIN, USER). Esto se implementa a través de anotaciones como `@PreAuthorize` y configuraciones en las rutas.

### - Filtros de seguridad por URL

Se han definido permisos granulares a nivel de URL, permitiendo acceso público solo a ciertos recursos y protegiendo el resto bajo autenticación y autorización.

### - Validación de permisos a nivel de método

En operaciones críticas del backend se utilizan anotaciones como `@PreAuthorize("hasRole('ADMIN')")` para garantizar que únicamente usuarios autorizados puedan ejecutar dichas acciones.

### - Protección CSRF activa

La protección CSRF está habilitada globalmente, salvo en endpoints explícitamente definidos como seguros para no requerirla (por ejemplo, APIs REST externas).

## Protección de Datos:

### - Aislamiento multi-tenant

El sistema soporta arquitectura multi-inquilino gracias a la clase `CustomRoutingDataSource`, que selecciona dinámicamente la base de datos según el tenant actual, asegurando así el aislamiento de datos entre clientes.

### - Validación de inputs

Todos los campos recibidos desde el frontend son validados antes de ser procesados, mitigando riesgos como inyección SQL o XSS.

### - Registro (logging) de accesos y cambios

Se registran todos los eventos relevantes como logins exitosos/fallidos, cambios en configuraciones críticas y accesos a recursos sensibles, facilitando auditorías futuras.

## Limitación de Tasa (Rate Limiting)

### - Protección de rutas POST críticas con Bucket4J

Para prevenir ataques de fuerza bruta y abuso de servicios públicos, se ha implementado una política de limitación de intentos por dirección IP en las siguientes rutas:

- /login

- /registro

- /enviar-contacto

- Configuración del límite de intentos

rate.limit.routes=/login,/registro,/enviar-contacto

rate.limit.max-attempts=5

rate.limit.refill-duration=60

Esto significa que si una IP realiza más de 5 solicitudes en 60 segundos, será bloqueada temporalmente hasta que se reinicie su cuota.

- Filtro personalizado: LoginRateLimitingFilter

Este filtro intercepta las peticiones a las rutas protegidas y aplica la lógica de control de tasa, devolviendo una respuesta HTTP 429 (Too Many Requests) cuando se supera el umbral permitido.

Integración en Spring Security:

El sistema de seguridad está completamente integrado con Spring Security mediante la clase SecurityConfig, donde se define:

- Configuración de URLs públicas y privadas.
- Personalización del formulario de login.
- Manejo del logout, incluyendo limpieza de cookies y reseteo de contexto de datos.
- Inclusión del filtro personalizado de rate limiting antes del filtro estándar de autenticación (UsernamePasswordAuthenticationFilter).
- Ignorar protección CSRF en endpoints específicos como APIs REST o reportes.

## 5. TECNOLOGÍAS UTILIZADAS

### 5.1 Backend: Spring Boot

Se ha utilizado Spring Boot 3.3.5 como framework principal del backend, lo que permite desarrollar aplicaciones empresariales en Java con una configuración mínima y una estructura modular. Entre sus características destacadas se encuentran:

- Autoconfiguración inteligente mediante convenciones.
- Servidor embebido Tomcat que facilita el despliegue sin necesidad de contenedores externos.
- Gestión optimizada de dependencias a través del spring-boot-starter-parent.
- Soporte para múltiples perfiles de entorno (dev, prod), lo que permite una configuración flexible.

Módulos de Spring empleados:

- Spring MVC: Implementación de controladores web y APIs REST.
- Spring Security 6: Gestión de autenticación y autorización basada en roles.
- Spring Data JPA: Acceso a datos mediante repositorios sobre entidades persistentes.
- Spring Boot Actuator: Exposición de métricas, salud y trazabilidad.
- Spring Boot Mail: Envío de correos automáticos mediante SMTP.

- Spring Cache + Caffeine: Mejora del rendimiento mediante almacenamiento en memoria.

El proyecto utiliza Java 17 como versión del JDK, garantizando compatibilidad con las últimas funcionalidades del lenguaje.

## 5.2 Frontend: Thymeleaf + Bootstrap

El motor de plantillas utilizado es Thymeleaf 3.1.x, integrado con Spring Boot para la generación de contenido HTML desde el servidor. Permite una sintaxis natural basada en HTML5 y una estructura modular mediante fragmentos reutilizables. También ofrece compatibilidad con Spring Security para mostrar o ocultar contenido dinámico según los permisos del usuario.

Para el diseño visual se ha empleado Bootstrap 5.3.x, que proporciona:

- Un sistema de rejilla responsiva con enfoque mobile-first.
- Componentes predefinidos como formularios, botones, alertas y navegación.
- Estilos personalizables y adaptación al branding del sistema.
- Iconografía mediante Bootstrap Icons.

JavaScript nativo ha sido utilizado para complementar la experiencia del usuario, implementando validaciones en tiempo real, peticiones AJAX y animaciones con la librería AOS (Animate On Scroll).

## 5.3 Persistencia de Datos: MySQL y Multitenancy

La base de datos principal del sistema es MySQL 8.0+, por su alto rendimiento en aplicaciones web, compatibilidad con MariaDB y facilidad de administración. Se han aplicado buenas prácticas de diseño de esquemas y uso de índices para mejorar el acceso concurrente.

El sistema implementa una arquitectura multi-tenant con las siguientes características:

- Separación de datos por organización mediante bases de datos independientes.
- Conmutación dinámica de fuentes de datos (DataSource) usando un AbstractRoutingDataSource.
- HikariCP como gestor de conexiones de alto rendimiento.
- Gestión automática de transacciones y aislamiento de sesiones.

## 5.4 Generación de Informes: JasperReports

Para la generación de reportes se ha integrado JasperReports 7.0.2, un motor potente para la creación de informes empresariales.

Se utiliza JasperSoft Studio como entorno de diseño visual, con soporte para:

- Exportación en múltiples formatos: PDF, Excel, HTML.
- Visualizaciones mediante gráficos, tablas dinámicas y componentes personalizados.
- Integración directa con Spring Boot mediante servicios y beans configurados.

Funcionalidades clave implementadas:

- Informes detallados por cliente, campaña y comercial.
- Métricas sobre rendimiento comercial, interacción y conversión.
- Dashboards interactivos con filtros y exportación.

- Automatización de reportes mediante programación periódica.

### 5.5 Control de Versiones: Git

El control de versiones se gestiona mediante Git, siguiendo el modelo de ramas Git Flow, que permite un desarrollo organizado y colaborativo.

Las ramas utilizadas son:

- main: Rama principal con el código listo para producción.
- develop: Rama de integración de nuevas funcionalidades.
- feature/\*: Ramas específicas para cada nueva característica.

Esta metodología permite mantener una trazabilidad clara del proyecto, favorece la colaboración en equipo y reduce errores en el proceso de despliegue.

## 6. PLANIFICACIÓN TEMPORAL

### 6.1 Metodología Seleccionada

- Revisiones regulares cada 2 viernes con el tutor y entre los desarrolladores
- Entrega continua con subida a feature → develop constante.

### 6.2 Fases del Desarrollo

Fase 1: Investigación y Análisis (4 semanas)

- Estudio de tecnologías y frameworks
- Definición de requisitos detallados
- Diseño de arquitectura multi-tenant
- Configuración del entorno de desarrollo

Fase 2: Desarrollo Core (6 semanas)

- Implementación de la base multi-tenant
- Sistema de autenticación y autorización
- Modelos de datos y repositorios
- APIs básicas de gestión

Fase 3: Funcionalidades CRM (4 semanas)

- Gestión de campañas y clientes
- Sistema de interacciones
- Dashboard y métricas básicas
- Integración frontend-backend

Fase 4: Sistema de Reportes (3 semanas)

- Configuración de JasperReports
- Diseño de plantillas de reportes
- Generación automática de PDFs
- Exportación a múltiples formatos

Fase 5: Testing y Refinamiento (3 semanas)

- Pruebas unitarias e integración
- Testing de usabilidad
- Optimización de rendimiento con cacheo y librerías no bloqueantes
- Corrección de bugs

### 6.3 Recursos Necesarios

#### Equipo de Desarrollo

- Desarrolladores Full-Stack : Profesionales con experiencia en tecnologías de backend (Java, Spring Boot) y frontend (Thymeleaf, Bootstrap), responsables del desarrollo integral del sistema.

#### Herramientas y Software

##### Entornos de Desarrollo (IDE)

- IntelliJ IDEA : Utilizado para la escritura y depuración del código Java y configuración de Spring Boot.

##### Bases de Datos

- MySQL Workbench : Para modelado, gestión y pruebas sobre la base de datos.
- MariaDB : Alternativa compatible con MySQL utilizada en entornos locales o pruebas.

##### Diseño y Reporting

- JasperSoft Studio : Creación y edición visual de informes en JasperReports.
- Software de edición gráfica y textual (por ejemplo: GIMP, LibreOffice, Inkscape): Para recursos visuales y documentación.

##### Control de Versiones

- Git : Sistema de control de versiones distribuido.
- GitHub : Repositorio central y plataforma de colaboración.
- Fork Client : Cliente de escritorio para gestión visual de ramas y commits.

##### Infraestructura Cloud

- Amazon Web Services (AWS) :
  - EC2 (Elastic Compute Cloud) : Para el despliegue del servidor de aplicación (Spring Boot + Tomcat).
  - RDS (Relational Database Service) : Para alojar las bases de datos MySQL de producción, incluyendo el soporte multitenant.
  - Opcionalmente , se pueden utilizar servicios complementarios como:
    - S3 : Almacenamiento de archivos y respaldos.
    - CloudWatch : Monitorización del sistema en producción.

### 6.4 Costes del Proyecto

En este apartado presento una estimación sencilla de los costes del proyecto XCRM, basada en el tiempo que hemos dedicado y los recursos utilizados.

#### Recursos Humanos

El equipo está formado por tres desarrolladores, que hemos dedicado aproximadamente 40 horas cada uno al proyecto. Considerando un coste medio por hora de 20 €, el coste total por el trabajo del equipo sería:

- $3 \text{ personas} \times 40 \text{ horas} \times 20 \text{ €/hora} = \mathbf{2.400 \text{ €}}$

#### Materiales, Hardware y Costes Indirectos

Cada uno utilizamos nuestro propio equipo de trabajo, pero para cubrir consumibles, electricidad, conexión a internet y espacio de trabajo estimamos un coste aproximado de 140 € durante el mes de desarrollo.

#### Software e Infraestructura

Utilizamos principalmente software libre y gratuito, por lo que no hay costes de licencias. Sin embargo, el alojamiento en la nube (AWS) tiene un coste aproximado de 100 € durante el periodo de desarrollo.

---

#### Resumen Total

Concepto	Coste (€)
Recursos Humanos	2.400 €
Materiales, Hardware y Costes Indirectos	140 €
Software e Infraestructura	100 €
<b>TOTAL</b>	<b>2.640 €</b>

## 7. DISEÑO E IMPLEMENTACIÓN

### 7.1 Test Unitarios

#### Objetivo de las Pruebas Unitarias

Las pruebas unitarias se han diseñado con el objetivo de verificar el correcto funcionamiento de la lógica de negocio encapsulada en las clases de servicio. Se han creado pruebas para cada uno de los servicios definidos en el paquete service/, utilizando JUnit 5 como framework de pruebas y Mockito para la simulación (mocking) de dependencias externas.

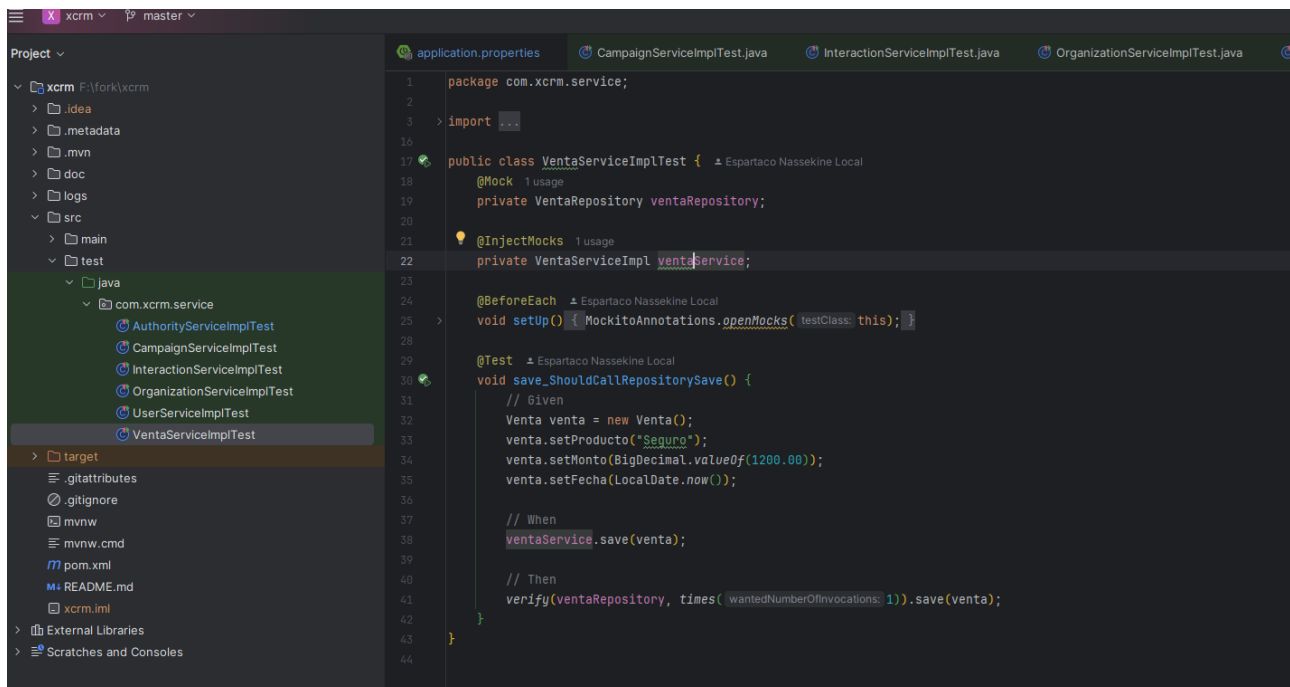
Se pretende tener el mayor porcentaje de código cubierto para así garantizar el correcto funcionamiento del programa y futuros cambios, que no rompan el actual.

#### Herramientas Utilizadas

- JUnit 5: para estructurar y ejecutar las pruebas.
- Mockito: para simular componentes como repositorios, EntityManager, y otros servicios inyectados, evitando dependencias reales.
- Spring Boot Test.

#### Estructura de los Tests

```
src/test/java/
├── com.xcrm.service/
│   ├── CampaignServiceImplTest.java
│   ├── ClientServiceImplTest.java
│   ├── ReportServiceImplTest.java
│   └── AuthorityServiceImplTest.java
```



The screenshot shows an IDE with the project structure on the left and the code of `VentaServiceImplTest.java` in the center. The project structure shows the following hierarchy:

```

xcrm
├── .idea
├── .metadata
├── .mvn
├── doc
├── logs
├── src
│   ├── main
│   └── test
│       └── java
│           └── com.xcrm.service
│               ├── AuthorityServiceImplTest
│               ├── CampaignServiceImplTest
│               ├── InteractionServiceImplTest
│               ├── OrganizationServiceImplTest
│               ├── UserServiceImplTest
│               └── VentaServiceImplTest
├── target
├── .gitattributes
├── .gitignore
├── mvnw
├── mvnw.cmd
├── pom.xml
├── README.md
├── xcrm.iml
└── External Libraries
  
```

The code in the center is the `VentaServiceImplTest` class, which is a JUnit 5 test class. It uses Mockito to mock the `VentaRepository` and the `VentaService` interface. The test class is annotated with `@Mock` and `@InjectMocks`. The `setUp()` method uses `MockitoAnnotations.openMocks(this)` to open the mocks. The `save_ShouldCallRepositorySave()` test method uses `Mockito.verify` to verify that the `save` method of `VentaService` is called with the correct arguments.

```

1 package com.xcrm.service;
2
3 import ...
4
5 public class VentaServiceImplTest {
6     @Mock
7     private VentaRepository ventaRepository;
8
9     @InjectMocks
10    private VentaServiceImpl ventaService;
11
12    @BeforeEach
13    void setUp() {
14        MockitoAnnotations.openMocks(this);
15    }
16
17    @Test
18    void save_ShouldCallRepositorySave() {
19        // Given
20        Venta venta = new Venta();
21        venta.setProducto("Seguro");
22        venta.setMonto(BigDecimal.valueOf(1200.00));
23        venta.setFecha(LocalDate.now());
24
25        // When
26        ventaService.save(venta);
27
28        // Then
29        verify(ventaRepository, times(wantedNumberOfInvocations(1))).save(venta);
30    }
31 }
  
```



## 7.2 Estructura del Código

Organización de Paquetes:

```
com.xcrm/
├── configuration/      # Configuraciones de Spring
│   ├── SecurityConfig.java
│   ├── MultiTenantConfig.java
│   └── JasperConfig.java
├── controller/
│   ├── web/           # Controladores MVC
│   │   ├── HomeController.java
│   │   ├── CampaignController.java
│   │   └── ReportController.java
│   └── rest/          # APIs REST
│       ├── CampaignRestController.java
│       └── MetricsRestController.java
├── dto/               # Objetos de transferencia
│   ├── CampaignDTO.java
│   ├── ClientDTO.java
│   └── InteractionDTO.java
├── model/             # Entidades JPA
│   ├── Organization.java
│   ├── Campaign.java
│   ├── Client.java
│   └── Interaction.java
├── repository/        # Repositorios de datos
│   ├── CampaignRepository.java
│   ├── ClientRepository.java
│   └── InteractionRepository.java
├── service/           # Lógica de negocio
│   ├── CampaignService.java
│   ├── ClientService.java
│   ├── ReportService.java
│   └── TenantService.java
└── utils/             # Utilidades
    ├── DateUtils.java
    ├── SecurityUtils.java
    └── TenantContext.java
```

## 7.3 Instalación y Configuración

Prerrequisitos del Sistema

Antes de instalar XCRM, asegúrate de tener:

Software Requerido:

- Java JDK 17 o superior
- MySQL 8.0 o MariaDB 10.6+
- Git (para clonar el repositorio)

Verificación de Java:

```
java -version
```

# Debe mostrar: openjdk version "17.0.x" o superior

Verificación de MySQL:

```
mysql --version
```

# Debe mostrar: mysql Ver 8.0.x o superior

### Paso 1: Clonar el Repositorio

# Clonar desde GitHub

```
git clone https://github.com/True1Santony/xcrm.git
cd xcrm
```

# Cambiar a la rama de desarrollo (opcional)

```
git checkout develop
```

### Paso 2: Configuración de Base de Datos

Crear base de datos principal:

-- Conectarse a MySQL como root

```
mysql -u root -p
```

-- Crear la base de datos principal

```
CREATE DATABASE mi_app CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

-- Crear usuario para la aplicación (opcional pero recomendado)

```
CREATE USER 'xcrm_user'@'localhost' IDENTIFIED BY 'xcrm_password_2025';
```

```
GRANT ALL PRIVILEGES ON mi_app.* TO 'xcrm_user'@'localhost';
```

```
GRANT CREATE ON *.* TO 'xcrm_user'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

### Paso 3: Configuración de la Aplicación

Editar archivo de configuración:

# src/main/resources/application.properties

# Configuración de la base de datos principal

```
spring.datasource.url=jdbc:mysql://localhost:3306/mi_app?useSSL=false&serverTimezone=UTC
```

```
spring.datasource.username=xcrm_user
```

```
spring.datasource.password=xcrm_password_2025
```

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

# Configuración JPA

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.jpa.show-sql=false
```

```
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
```

# Configuración del servidor

```
server.port=8080
```

# Configuración de logging

```
logging.level.com.xcrm=DEBUG
```

```
logging.file.name=logs/xcrm.log
```

### Paso 4: Compilación y Ejecución

Usando Maven Wrapper (recomendado):

# En sistemas Unix/Linux/Mac

```
./mvnw clean install
```

```
./mvnw spring-boot:run
```

# En Windows

```
mvnw.cmd clean install
```

```
mvnw.cmd spring-boot:run
```

Usando Maven instalado:

```
mvn clean install
mvn spring-boot:run
```

Paso 5: Verificación de la Instalación  
Acceder a la aplicación:

URL: <http://localhost:8080>

Página de inicio esperada:

- [intex.html](#)

## 7.4 API Desarrollada



### Actuator / Monitoring

Método	Ruta	Descripción
GET	/actuator	Muestra los endpoints disponibles
GET	/actuator/health	Estado de salud del sistema
GET	/actuator/health/**	Detalles extendidos del estado
GET	/actuator/mappings	Lista todas las rutas mapeadas
GET	/actuator/info	Información general del sistema



### Páginas Públicas

Método	Ruta	Descripción
GET	/	Página principal
GET	/caracteristicas	Características del producto
GET	/precios	Información sobre precios
GET	/contacto	Formulario de contacto
POST	/enviar-contacto	Procesa el formulario de contacto
GET	/aviso-legal	Aviso legal del sitio
GET	/login	Formulario de inicio de sesión
GET	/registro	Formulario de registro
POST	/registro	Procesa el registro de nuevos usuarios



### Gestión de Usuarios

Método	Ruta	Descripción
GET	/usuarios/administration	Panel de administración de usuarios
GET	/usuarios/configuracion	Configuración del usuario actual
GET	/usuarios/usuarios/configuracion	Configuración avanzada de usuarios
POST	/usuarios/actualizar-configuracion	Actualiza configuraciones del usuario
POST	/usuarios/registrarse	Registro de nuevo usuario
POST	/usuarios/editar	Edita datos de un usuario existente
POST	/usuarios/eliminar	Elimina un usuario del sistema
GET	/usuarios/edit-mi-perfil	Edición del perfil del usuario actual
POST	/usuarios/edit-mi-perfil/update	Guarda cambios del perfil del usuario
POST	/usuarios/guardar-foto	Sube o actualiza foto de perfil
POST	/usuarios/enviar-ticket-edicion	Envía ticket para edición de datos



## Campañas (Campaigns)

Método	Ruta	Descripción
GET	/campaigns/administration	Panel de administración de campañas
GET	/campaigns/edit/{id}	Formulario de edición de campaña
POST	/campaigns/update	Actualiza los datos de una campaña
POST	/campaigns/delete	Elimina una campaña
POST	/campaigns	Crea una nueva campaña



## Ventas / Comercial

Método	Ruta	Descripción
GET	/sales	Dashboard de ventas/comercial
GET	/sales/clientByIdCampaign	Cientes asociados a una campaña
GET	/sales/interaction/{id}	Formulario para registrar interacción
POST	/sales/interaction/save	Guarda una interacción registrada
POST	/sales/assign	Asigna campañas a usuarios comerciales



## Reportes

Método	Ruta	Descripción
GET	/report	Dashboard de reportes
GET	/report/admin/ventas	Reporte de ventas por comercial
GET	/report/admin/interaccionesPorVenta	Reporte de interacciones por venta
POST	/report/generate	Genera un informe personalizado



## Clientes

Método	Ruta	Descripción
GET	/clients	Dashboard de gestión de clientes
POST	/clients	Registra un nuevo cliente
GET	/clients/edit	Muestra formulario de edición de cliente
POST	/clients/update	Actualiza los datos de un cliente
POST	/clients/delete	Elimina un cliente



## Configuración y Administración

Método	Ruta	Descripción
POST	/api/cache/clear	Limpia la caché del sistema
GET	/api/config/export-db	Exporta la base de datos del sistema



## API REST - Reportes

Método	Ruta	Descripción
GET	/api/ventas-por-comercial	Datos de ventas agrupados por comercial
GET	/api/ventas-por-campania	Datos de ventas agrupados por campaña
GET	/api/interacciones-por-venta	Datos de interacciones por venta

## 🚫 Errores y Recursos Estáticos

Método	Ruta	Descripción
Todos	/error	Manejador genérico de errores HTTP
GET	/webjars/**	Acceso a recursos estáticos (WebJars)
GET	/**	Acceso a archivos estáticos generales
GET	/uploads/**	Archivos subidos por usuarios (ej: imágenes)

## 8. MANUAL DEL USUARIO

### Capturas y funcionalidades (descripción)

#### 8.1 Registro y login

### REGISTRO

Selecciona un Plan:

-- Elige un Plan --

Registrar

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

### Inicio de Sesión

Usuario

Contraseña

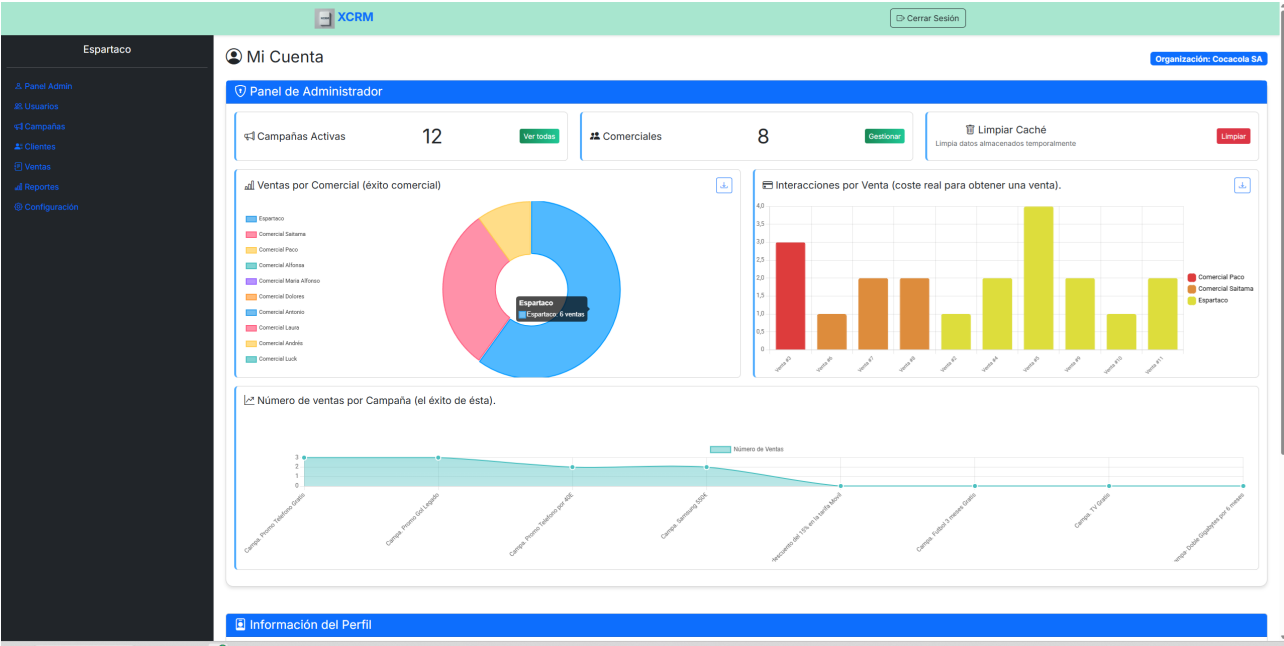
Iniciar Sesión

¿No tienes cuenta? [Regístrate aquí](#)

[Volver a Inicio](#)

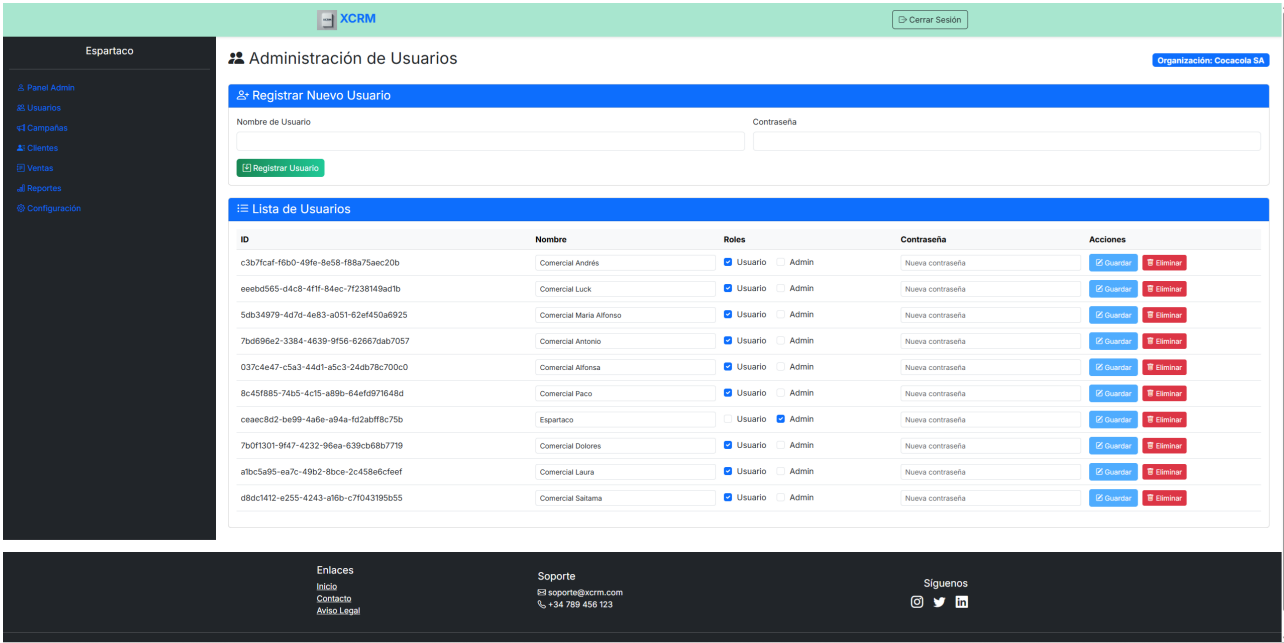
La organización puede registrarse o logearse un usuario. Si es registro, la app crea una base de datos para la organización, sus tablas y registra todo en la DB Getaway y la propia DB. Si es login, la app busca al usuario en la DB Central y establece la session/contexto se su organización.

8.2 Dashboard del usuario



Tras el login, si el usuario es ADMIN\_USER, vera un resumen de métricas de su organización, la actividad comercial, exitos de campañas, costes operacionales... Se calculan diariamente, si no se limpia el cache.

8.3 Administración de usuarios



Se puede registrar nuevos usuarios y editarlos. Se agregan o editan tanto en la base de datos de la organización como la central de la app.

8.4 Administración de campañas

Espartaco

Panel Admin

Usuarios

Campañas

Cientes

Ventas

Reportes

Configuración

Administración de Campañas

Organización: CocaCola SA

Crear Nueva Campaña

Nombre

Descripción

Fecha de Inicio

Fecha de Fin

Seleccionar Clientes

Crear Campaña

Lista de Campañas

ID	Nombre	Descripción	Fecha Inicio	Fecha Fin	Cientes	Comerciales	Acciones
1	Campa. Promo Telefono Gratis	Se trata de llamar a los clientes para ofrecer un movil gratis por una permanencia de 3 Años	04/04/2025	30/04/2025	1 cliente(s)	Comercial Alfonso, Comercial Pascu, Espartaco,	No editable Eliminar
2	Campa. Promo Telefono por 40€	Se trata de llamar a los clientes para ofrecer un movil gratis por una permanencia de 3 Años	04/04/2025	30/04/2025	1 cliente(s)	Comercial Alfonso, Espartaco, Comercial Laura,	No editable Eliminar
3	Campa. Promo descuento del 15% en la tarifa Movil	ofrecer el descuento en la tarifa movil e yntentar cerrar una venta paralela, contenido de tv, fibra optica, o lo que tu quieras	04/04/2025	05/06/2029	1 cliente(s)	Comercial Lock, Comercial Maria Alfonso, Comercial Pato,	No editable Eliminar
4	Campa. Futbol 3 meses Gratis	Se ofrece 3 meses gratis el futbol a los nuevos clientes asi que llamar a los de la competencia	18/04/2025	30/04/2025	1 cliente(s)	Comercial Maria Alfonso, Comercial Laura,	No editable Eliminar
5	Campa. Samsung 550€	Ofrecer a todos el telefono financiado a 3 años	18/04/2025	30/04/2025	1 cliente(s)	Comercial Andria, Espartaco, Comercial Dolores,	No editable Eliminar
8	Campa. TV Gratis	Enviamos una tv gratis si contrata futbol por 1 año	03/05/2025	12/05/2025	1 cliente(s)	Espartaco, Comercial Dolores, Comercial Salma,	No editable Eliminar
9	Campa. Promo Gol L Leado	Empezara el mes que lueves	01/06/2025	30/06/2025	Ninguno	Comercial Andria, Espartaco,	No editable Eliminar

Si es admin, puede crear nuevas campañas, eliminarlas o editarlas (si la fecha lo permite). Ademas se visualizan todas las campañas con los comerciales que están suscritos a estas y los clientes.

8.5 Administración de clientes

Espartaco

Panel Admin

Usuarios

Campañas

Cientes

Ventas

Reportes

Configuración

Administración de Clientes

Organización: CocaCola SA

Añadir Cliente

Nombre

Telefono

Email

Dirección

Campañas

Comerciales

Guardar Cliente

Lista de Clientes

ID	Nombre	Dirección	Telefono	Email	Campañas	Comerciales	Acciones
5	Cliente Editado	Calle Fondón 1, Bajo C	624584589	espartaconassekine@gmail.com	2 campaña(s)	3 comercial(es)	Editar Eliminar
6	Maria Blanco Martinez	Calle Paseo de la Habana 25, Bajo C, 28033, Madrid	658568598	cliente2@gmail.com	1 campaña(s)	3 comercial(es)	Editar Eliminar
7	Manolito Perez Hernandez	Calle Eduardo Chillida, 12, 1º 2, Parla Madrid	+ 34 625845874	manlito625@gmail.com	1 campaña(s)	1 comercial(es)	Editar Eliminar
8	Nacisa Macarena	Calle Galeo 42, 6º C, Galicia 48000	698587458	narcisa@gmail.com	3 campaña(s)	1 comercial(es)	Editar Eliminar
9	Alejandro Blanco Polos	Calle Marco Polo, 7º C, Galicia 48020	649854878	ale125blanco@hotmailcon	1 campaña(s)	1 comercial(es)	Editar Eliminar
10	Jose Luis Gonzalez	Paseo de Recoletos 250, 15º 1, 28005 Madrid	665864950	jose554456@gmail.com	1 campaña(s)	1 comercial(es)	Editar Eliminar
11	Laura Méndez	Calle Mayor 12, Madrid	612345678	laura.mendez@gmail.com	8 campaña(s)	Ninguno	Editar Eliminar
12	José Ramirez	Avenida de Andalucía 45, Sevilla	699887766	jose.ramirez@yahoo.es	Ninguna	Ninguno	Editar Eliminar

Si es admin, puede añadir clientes, al mismo tiempo, opcionalmente, suscribir a campañas o/y comerciales. Se visualizan todos los clientes y a que campañas están suscritos, se les puede editar y eliminar.

8.6 Edición del cliente

Espartaco

Panel Admin

Usuarios

Campañas

Cientes

Ventas

Reportes

Configuración

Inicio

Cerrar Sesión

Editar Cliente

Volver a la lista

Editar Cliente ID: 5

Nombre

Ciente Editado

Telefono

624584589

Email

espartaconaskekine@gmail.com

Dirección

Calle Fondón 1, Bajo C

Campañas

Campa. Promo Telefono Gratis

Campa. Promo Telefono por 40€

Campa. Promo descuento del 15% en la tarifa Movil

Campa. Futbol 3 meses Gratis

Campa. Samsung 550€

Comerciales

Comercial Alfonso

Comercial Maria Alfonso

Comercial Dolores

Comercial Antonio

Comercial Pico

Mantén Ctrl (Windows) o Command (Mac) para seleccionar múltiples

Seleccionados

Campa. Doble Display por 6 meses

Campa. Promo Telefono por 40€

Guardar Cambios

Cancelar

Mantén Ctrl (Windows) o Command (Mac) para seleccionar múltiples

Seleccionados

Comercial Andes

Comercial Pico

Espartaco

Total edición del cliente.

8.7 Dashboard de ventas

Espartaco

Panel Admin

Usuarios

Campañas

Cientes

Ventas

Reportes

Configuración

Inicio

Cerrar Sesión

Dashboard de Ventas

Organización: CocaCola SA

Asignar a Comerciales Campañas

Comercial

Comercial Alfonso

Campañas

Campa. Promo Telefono Gratis

Campa. Promo Telefono por 40€

Campa. Promo descuento del 15% en la tarifa Movil

Campa. Futbol 3 meses Gratis

Asignar Campañas

Mis Campañas

Selecciona una campaña

Elige una campaña

Cientes de la campaña: Campa. Promo Telefono Gratis

Nombre	Telefono	Email	Estado última interacción	Fecha última interacción	Fecha venta	Acciones
Maria Blanco Martinez	658568598	cliente2@gmail.com	REALIZADA	07/05/2025 14:33	07/05/2025 00:00	Interactuar
Nacisa Macarena	698587458	narcisa@gmail.com	CERRADO	14/05/2025 18:14	14/05/2025 00:00	Interactuar
Alejandro Blanco Polos	649854878	ale125blanco@hotmail.com	Sin interacciones	N/A	Sin venta	Interactuar
Laura Méndez	612345678	laura.mendez@gmail.com	CERRADO	14/05/2025 10:19	14/05/2025 00:00	Interactuar
Jose Luis Amor	648578985	joseamor@gmail.com	Sin interacciones	N/A	Sin venta	Interactuar

Enlaces

Inicio

Contacto

Aviso Legal

Soporte

soporte@xcrm.com

+34 789 456 123

Síguenos

© 2024 XCRM. Todos los derechos reservados.

Si es admin, puede asignar a los comerciales las campañas, que luego se reflejaran en administración de campañas. Esto sirve para que cuando un comercial se logee se encuentre con sus campañas asignadas. Se escoge la campaña asignada a tratar, muestra los clientes asignados. Cada cliente, muestra la ultima interacción con éste y por el usuario activo (porque puede haber otros usuarios que hayan interactuado con este cliente) y el botón de comenzar a interactuar.



## 8.8 Interacciones

XCRM

Cerrar Sesión

Volver sin interactuar

Interacciones con Cliente

Alejandro Blanco Polos

Teléfono: 648854878

Email: ale125blanco@hotmail.com

Dirección: Calle Marco Polo, 7º C, Galicia 48020

Nueva Interacción

Tipo de Interacción

Visita

Estado

Cerrado

Observaciones

Se lo lleva

Marcar como venta realizada

Guardar Interacción

Llamar

Historial de Interacciones

Fecha	Tipo	Estado	Observaciones	Venta	Campaña	Trasado por
14/05/2025 10:29	LLAMADA	PENDIENTE	Le vuelvo a llamar dice que mañana se lo pregunta a su mujer	No	Campa. Promo Telefono por 40E	Espartaco
14/05/2025 10:30	VISITA	CERRADO	Se lleva el telefono	Si - 14/05/2025	Campa. Promo Telefono por 40E	Espartaco
14/05/2025 10:28	LLAMADA	PENDIENTE	le llamo por si le interesa el telefono, no lo sabe contrreto llamada para mas adelante	No	Campa. Promo Telefono por 40E	Espartaco
14/05/2025 10:30	EMAIL	EN_PROCESO	le envio por email todas las caracteristicas del telefono y las condiciones	No	Campa. Promo Telefono por 40E	Espartaco

Enlaces

Aquí se pretende registrar lo que se haya hecho con el cliente. Que tipo de interaccion es (visita, llamada, envío de email, whatsapp...), cual es el estado del cliente en esta interacción. Se pueden agregar comentarios para futuras iteraciones, marcar como finalizada, vendido, pendiente... Además se muestra el histórico total del cliente, todas las interacciones que ha habido con este independientemente de campaña o comercial. Estas interacciones sirven para realizar las métricas y reportes.

## 8.9 Generador de reportes

XCRM

Cerrar Sesión

Espartaco

Panel Admin

Usuarios

Campañas

Clientes

Ventas

Reportes

Configuración

Generador de Reportes

Organización: CocaCola SA

Tipo de Reporte

Todas las interacciones

Formato

Excel

Generar Reporte

Enlaces

Inicio

Contacto

Aviso Legal

Soporte

es soporte@xcrm.com

+34 789 456 123

Síguenos

© 2024 XCRM. Todos los derechos reservados.

Gracias a plantillas generadas previamente, en este apartado se pueden generar los reportes de todos los comerciales → campañas → clientes → interacciones, usuarios de la plataforma y todos los clientes. Están cacheados, se generan cada día, a no ser que se refresque desde dashboard del admin.

Admite 3 formatos, PDF, EXCEL y HTML.

## 8.1.0 Formulario de Contacto con opción de adjuntos (Dropbox + local)

 **XCRM** ★ Características ✉ Contacto

[Mi Cuenta](#) [Cerrar Sesión](#)

# ¿Tienes alguna pregunta?

Nos encantaría saber de ti. Completa el siguiente formulario y nos pondremos en contacto contigo lo antes posible.

Nombre

Correo Electrónico

Asunto

Mensaje

Buenas tardes,  
me gustaria tener una reunión para explicaros mi propuesta al proyecto.  
Adjunto la propuesta  
Saludo Cordial

Adjuntar archivo desde  Dropbox

[Selecccionar archivo](#)  1.png [Cancelar](#)

Adjuntar archivo desde el ordenador

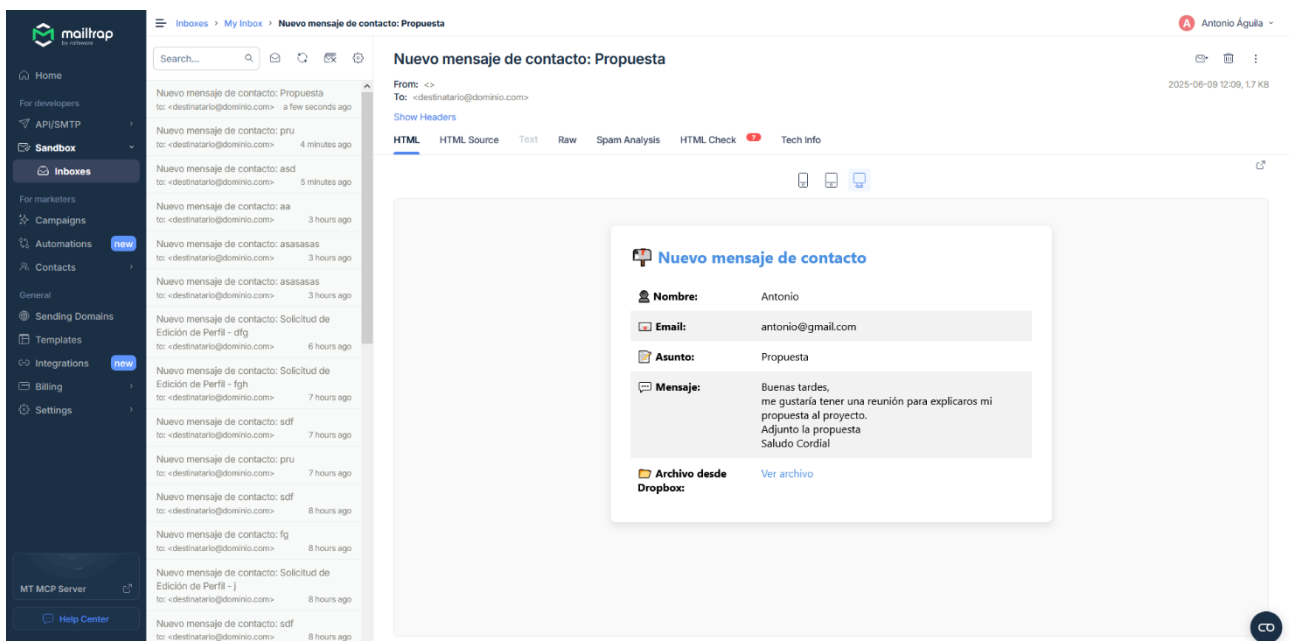
[Examinar...](#) No se ha seleccionado ningún archivo.

Tamaño máximo: 4 MB. Formatos permitidos: PDF, Word, JPG, PNG.

[Enviar](#)

Este formulario permite que cualquier usuario nos envíe un mensaje con sus datos y, si lo desea, puede adjuntar archivos desde su ordenador o directamente desde Dropbox. Está integrado con Bootstrap y se comunica con el backend usando un POST a /enviar-contacto

Respuesta:



### 8.1.1 Gestión de campañas (Crear, editar, eliminar y listar)

The screenshot displays the XCRM web application interface. At the top, a green header bar contains the XCRM logo and a hamburger menu icon. Below this, a dark sidebar on the left lists navigation options: 'Panel Admin', 'Usuarios', 'Campañas', 'Clientes', 'Ventas', 'Reportes', and 'Configuración'. The main content area has a title 'Administración de Campañas' and a button 'Organización: trump'. The specific page is 'Editar Campaña', with a 'Volver al listado' button. The form for 'Editar Campaña ID: 15' includes fields for 'Nombre' (Clientes Estandar), 'Descripción' (Descuentos), 'Fecha de Inicio' (15/09/2026), and 'Fecha de Fin' (15/11/2026). A 'Clientes Asociados' section shows a list with 'Oscar' and 'Vicente', where 'Oscar' is selected. A note at the bottom of the list says 'Mantén Ctrl (Windows) o Command (Mac) para seleccionar múltiples'. At the bottom of the form are 'Guardar Cambios' and 'Cancelar' buttons.

anto

Administración de Campañas

Organización: trump

anto

Editar Campaña

← Volver al listado

Editar Campaña ID: 15

Nombre	Descripción
Clientes Estandar	Descuentos
Fecha de Inicio	Fecha de Fin
15 / 09 / 2026	15 / 11 / 2026

Clientes Asociados

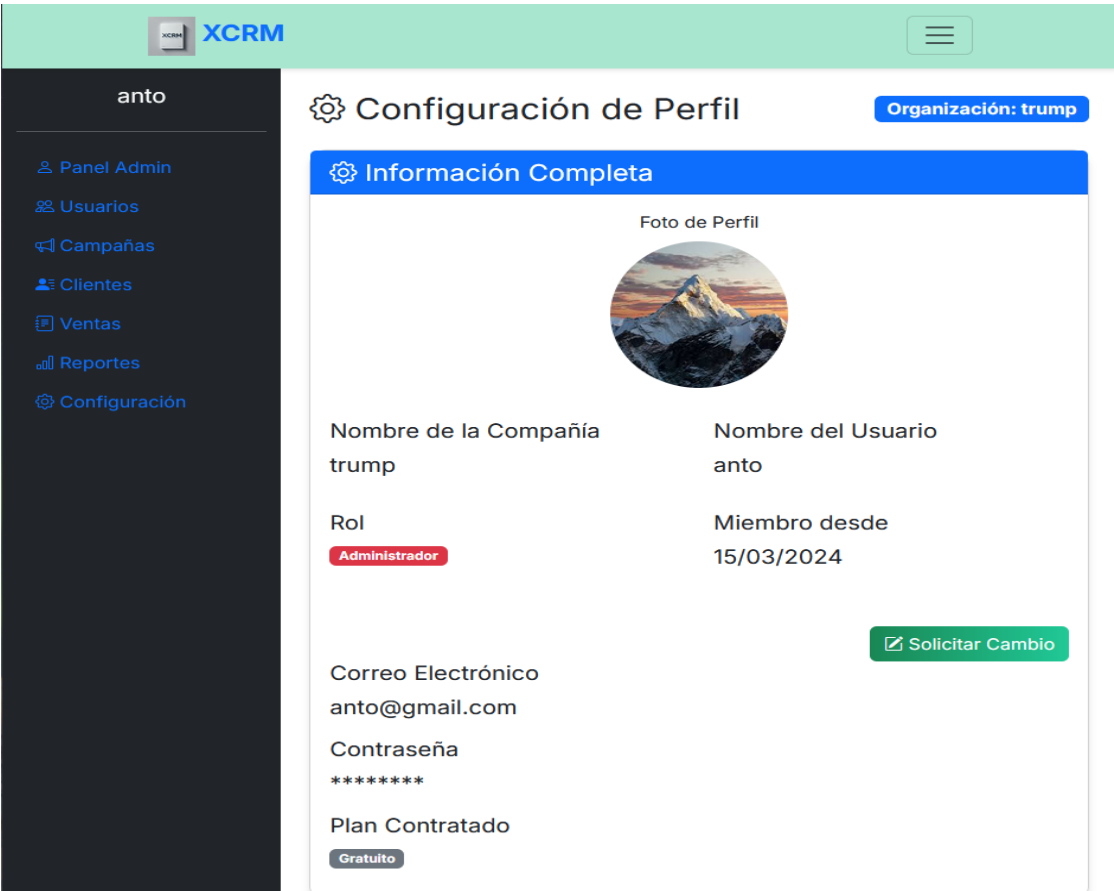
- Oscar
- Vicente

Mantén Ctrl (Windows) o Command (Mac) para seleccionar múltiples

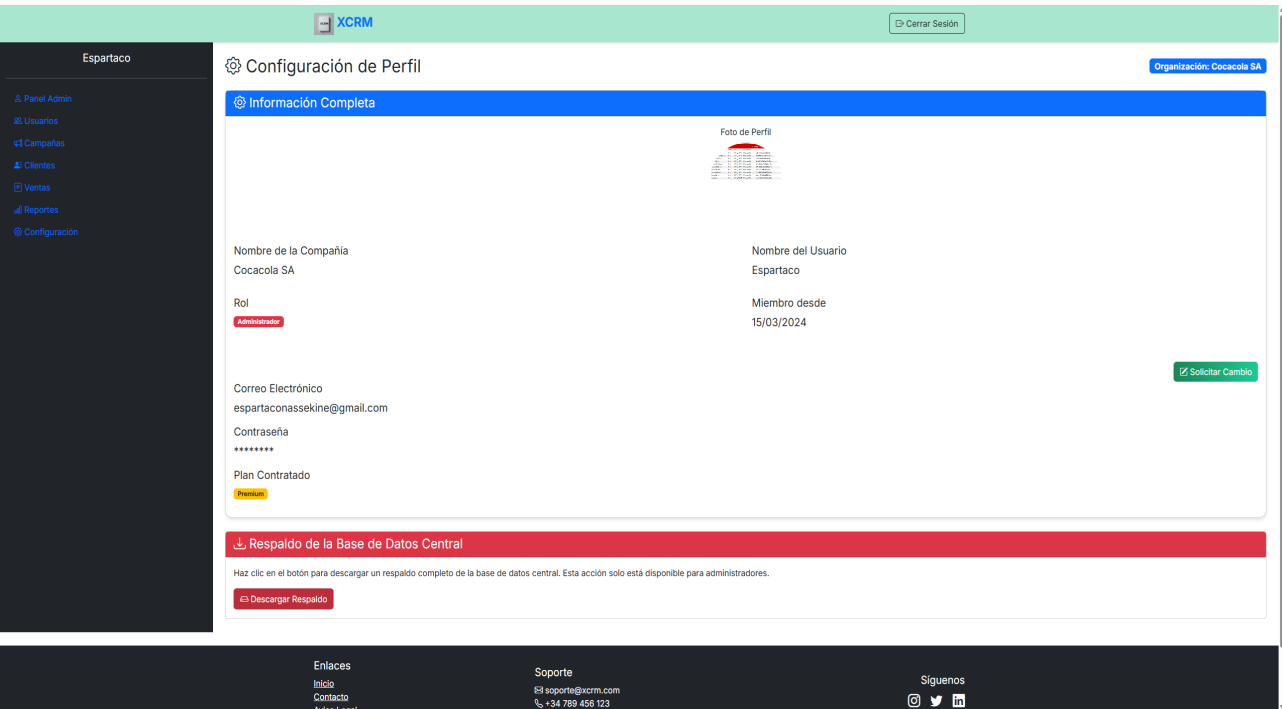
Guardar Cambios Cancelar

Este módulo permite gestionar campañas desde un panel administrativo. Se pueden crear campañas nuevas, editarlas (si aún no han comenzado), eliminarlas y ver un listado general. Además, se validan fechas y se asocian clientes a cada campaña.


8.1.2 Configuración y visualización del perfil de usuario con menú lateral y respaldo de base de datos para administradores




Esta página muestra los datos completos del perfil del usuario actual, incluyendo foto, organización, rol y plan contratado. Además, ofrece un menú lateral para navegar por las secciones principales del sistema y permite a los administradores descargar un respaldo de la base de datos.



### 8.1.3 Edición de perfil de usuario administrador

XCRM



Usuarios

Campañas

Clientes


Ventas

Reportes

Configuración

Información de Usuario

Foto de Perfil



Examinar...

No se ha seleccionado ningún archivo.

Nombre de la compañía

Rpm

Nombre del usuario

Antonio

Correo electrónico

Antonio@gmail.com

Selecciona un Plan:

Standard

Motivo de la solicitud

Quiero elegir el siguiente plan y cambiar los datos

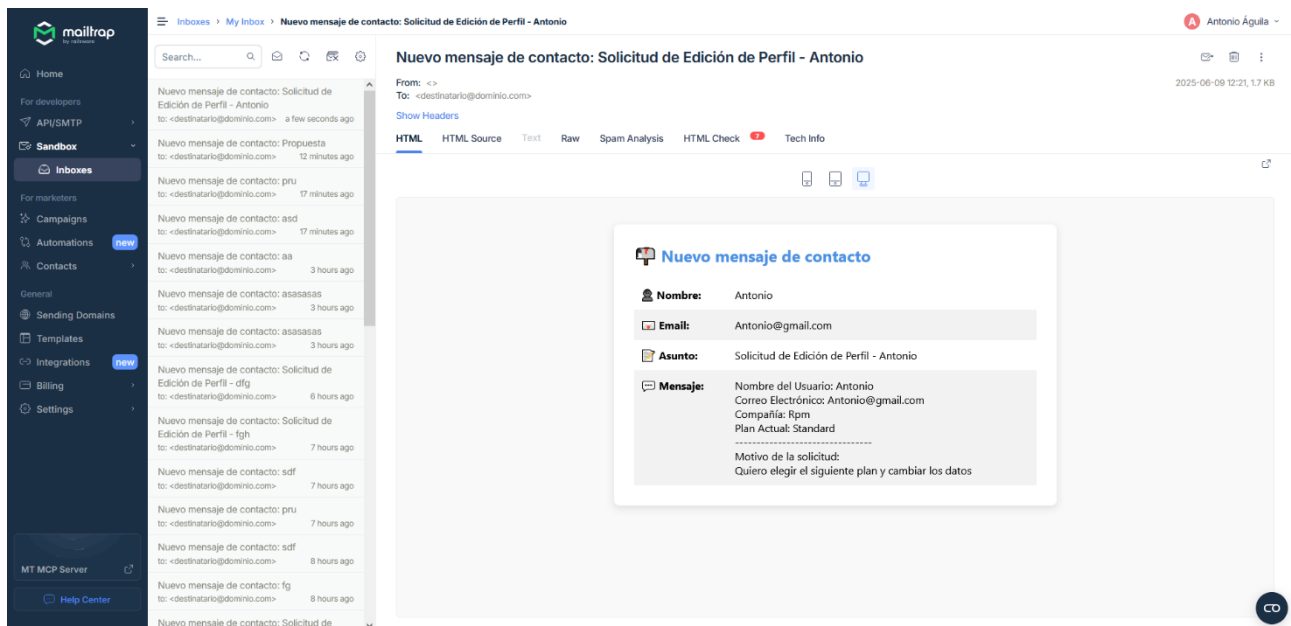
Guardar Foto

Enviar Ticket

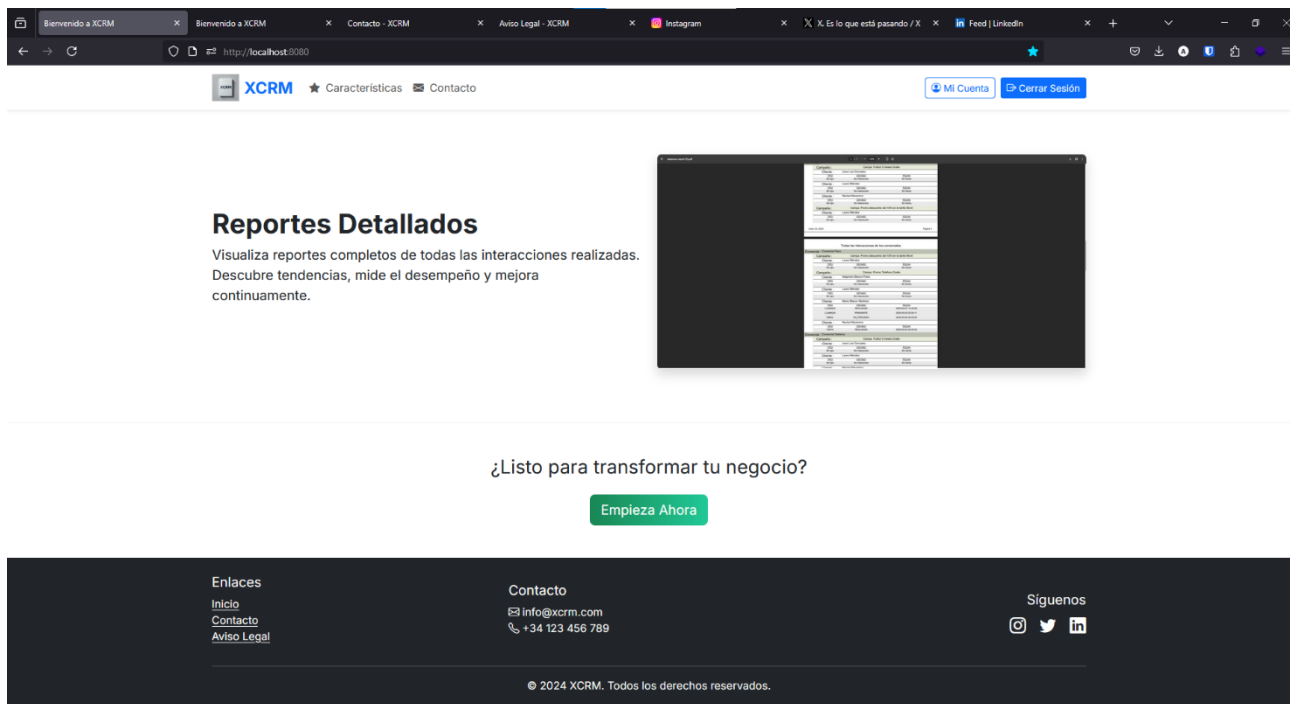
Cancelar

Esta pantalla permite al usuario administrador actualizar su foto de perfil y enviar una solicitud para modificar sus datos personales, como el nombre, correo o plan actual. Todo está integrado con formularios que se procesan en el backend y muestran mensajes de éxito o error según la acción realizada.

Respuesta:



### 8.1.4 Footer



Estas funcionalidades juntas conforman la estructura legal y de navegación básica del sitio web. La página de Aviso Legal ofrece a los usuarios toda la información necesaria para entender los términos, condiciones y políticas de privacidad del servicio. Los footers reutilizables garantizan una experiencia consistente y accesible en todo el sitio, diferenciando claramente entre la vista pública y la sección administrativa al incluir enlaces, información de contacto o soporte, y acceso a redes sociales.\_

## 9. CONCLUSIONES

El desarrollo del sistema XCRM ha permitido implementar una solución CRM multi-tenant moderna, segura y escalable, orientada a satisfacer las necesidades de organizaciones que gestionan campañas comerciales, interacciones y ventas. A lo largo del proyecto se han abordado desafíos técnicos significativos como la arquitectura multi-tenant basada en bases de datos por organización, la integración de JasperReports para generación de informes, y la configuración de seguridad con Spring Security, obteniendo resultados satisfactorios.

Entre los principales logros destacan:

- **Aislamiento total de datos entre organizaciones**, permitiendo cumplir con normativas de privacidad y facilitar tareas de mantenimiento como copias de seguridad por tenant.
- **Automatización en la creación de organizaciones y bases de datos**, reduciendo la fricción en el alta de nuevos clientes.
- **Generación de informes en múltiples formatos**, mejorando el análisis de rendimiento para toma de decisiones empresariales.
- **Interfaz amigable y responsiva** con Bootstrap y Thymeleaf, favoreciendo la experiencia de usuario.
- **Seguridad reforzada**, con autenticación encriptada, control de sesiones y limitación de intentos para rutas críticas.

El proyecto ha seguido una planificación ordenada por fases, cumpliendo los plazos estimados y validando cada módulo mediante pruebas unitarias y de integración. Gracias al uso de tecnologías robustas y al enfoque modular, XCRM se encuentra preparado para escalar y adaptarse a entornos reales.

Como mejora futura, se propone:

- Incorporar un sistema de notificaciones SMS.
- Permitir subir documentación de la organización, como explicación detallada de campañas, procedimientos.
- Importación/exportación de db mas detallada, no globalmente.
- Añadir un módulo de análisis predictivo con aprendizaje automático.
- Chatbot LLM, por API IA.
- Desplegar versiones móviles nativas para Android/iOS.
- Mejora de Arquitectura a Micro servicios distribuidos, si no fuera suficiente el escalado vertical.

XCRM no solo cumple los requisitos funcionales y no funcionales establecidos, sino que sienta una base sólida para la evolución futura del sistema.

## 10. BIBLIOGRAFÍA

### Frontend (HTML, CSS, JS, Bootstrap, Thymeleaf)

- HTML Living Standard. <https://html.spec.whatwg.org/>
- CSS – Cascading Style Sheets. <https://developer.mozilla.org/en-US/docs/Web/CSS>
- JavaScript Guide – MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- Bootstrap Documentation. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- Thymeleaf Official Guide. <https://www.thymeleaf.org/documentation.html>
- Plantillas reutilizables (th:replace, fragments) explicadas. <https://www.baeldung.com/thymeleaf-layouts>
- UI/UX Design Principles. <https://www.smashingmagazine.com/guidelines-for-mobile-web-design/>
- Accesibilidad Web (WAI-ARIA). <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA>
- JavaScript Fetch API. [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API)
- Form Validation in HTML and JavaScript. [https://developer.mozilla.org/en-US/docs/Learn/Forms/Form\\_validation](https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation)
- JavaScript FormData API. <https://developer.mozilla.org/en-US/docs/Web/API/FormData>
- Bootstrap Forms. <https://getbootstrap.com/docs/5.3/forms/overview/>
- Bootstrap Toasts. <https://getbootstrap.com/docs/5.3/components/toasts/>
- Thymeleaf Forms + Spring Boot. <https://www.baeldung.com/spring-thymeleaf-form-handling>
- UX Writing: microtextos para formularios y botones. <https://uxdesign.cc/ux-writing-101-3ccdfb70de1>
- Animaciones y feedback visual para formularios. <https://animate.style/>
- Form UX – Best Practices. <https://uxdesign.cc/form-design-best-practices-ecab23f0c63f>

### Backend (Java, Spring Boot, MySQL, Hibernate, etc.)

- Java SE 17 Documentation. <https://docs.oracle.com/en/java/javase/17/>
- Spring Boot Official Documentation. <https://docs.spring.io/spring-boot/docs/current/reference/html/>
- MySQL 8.0 Reference Manual. <https://dev.mysql.com/doc/refman/8.0/en/>
- Hibernate ORM Documentation. <https://hibernate.org/orm/documentation/>
- Project Lombok – Features. <https://projectlombok.org/features/all>
- Bucket4j – Token Bucket Rate Limiter for Java. <https://github.com/vladimir-bukhtoyarov/bucket4j>
- Mockito – Mocking Framework for Unit Tests. <https://site.mockito.org/>
- JUnit 5 User Guide. <https://junit.org/junit5/docs/current/user-guide/>
- Guía de configuración application.properties + dependencias Maven. <https://www.baeldung.com/spring-boot-jdbc-hikari>
- Guía de relaciones en JPA. <https://www.baeldung.com/jpa-join-types>
- Validación con anotaciones (@Valid, @NotBlank, etc.). <https://www.baeldung.com/spring-boot-bean-validation>
- Uso de LocalDate, conversión y formato. <https://www.baeldung.com/java-date-to-localdate>
- Ejemplo con MultipartFile, guardar y servir imágenes. <https://atacomsian.com/blog/spring-boot-file-upload>
- Spring Mail con Mailtrap. <https://www.baeldung.com/spring-email>
- Spring Boot MultipartFile y correo con archivos adjuntos. <https://www.baeldung.com/spring-email-attachment>
- Spring Boot Security – Session Management. <https://www.baeldung.com/spring-security-session>
- Spring Boot: Diferenciar entre usuarios autenticados y no autenticados. <https://www.baeldung.com/spring-security-authentication>
- UUID Generation en Java. <https://www.baeldung.com/java-uuid>
- Manejo de errores personalizados en Spring. <https://www.baeldung.com/exception-handling-for-rest-with-spring>
- Spring Boot Custom Validation Annotation. <https://www.baeldung.com/spring-mvc-custom-validator>
- Java Mail Sender con Spring. <https://docs.spring.io/spring-framework/reference/integration/email.html>
- JPA OneToMany y ManyToMany con formularios. <https://www.baeldung.com/jpa-many-to-many>
- Subir archivos a Dropbox desde Java (Dropbox SDK). <https://dropbox.github.io/dropbox-sdk-java/api-docs/v3.0.x/>
- Validación de fechas con LocalDate. <https://www.baeldung.com/java-date-validation>

### Herramientas de Desarrollo

- IntelliJ IDEA Documentation. <https://www.jetbrains.com/idea/documentation/>
- Git – <https://git-scm.com/book/en/v2>
- Fork Git GUI Client. <https://git-fork.com/>



- Maven – Project Management and Comprehension Tool. <https://maven.apache.org/guides/index.html>
- Postman Learning Center. <https://learning.postman.com/>
- Flyway Database Migrations. <https://flywaydb.org/documentation>
- Docker Documentation. <https://docs.docker.com/>
- Markdown Guide – Basic Syntax. <https://www.markdownguide.org/basic-syntax/>
- JasperReports Library. <https://community.jaspersoft.com/project/jasperreports-library>
- OWASP Foundation – Top 10 Web Application Security Risks. <https://owasp.org/www-project-top-ten/>
- Dropbox for Developers (API Java). <https://www.dropbox.com/developers/documentation/java>
- Spring Boot Testing con MockMvc y WebMvcTest. <https://www.baeldung.com/spring-boot-testing>
- Swagger OpenAPI para documentar tu backend. <https://swagger.io/docs/specification/about/>
- Spring Boot Actuator para monitorear tu app. <https://docs.spring.io/spring-boot/docs/current/actuator-api/html/>
- Spring Boot Logging (logback, errores, eventos importantes). <https://www.baeldung.com/spring-boot-logging>
- Spring Boot DevTools para hot reload.  
<https://docs.spring.io/spring-boot/docs/current/reference/html/using.html#using.devtools>
- SonarQube para análisis de calidad del código. <https://docs.sonarsource.com/sonarqube/latest/>

## **Seguridad y Buenas Prácticas**

- Protección contra CSRF en Spring Boot. <https://www.baeldung.com/spring-security-csrf>
- Validar tipos MIME y extensiones en uploads. <https://www.baeldung.com/spring-file-upload-content-type>
- OWASP Secure Headers. <https://owasp.org/www-project-secure-headers/>
- Cómo implementar roles y permisos en Spring Security. <https://www.baeldung.com/role-and-privilege-for-spring-security-registration>

## **Emails y Formularios Profesionales**

- Email Templates con Thymeleaf. <https://www.baeldung.com/spring-email-thymeleaf>
- Email con HTML y diseño responsive. <https://templates.mailchimp.com/design/>
- Spring Mail HTML Template + Adjuntos. <https://atacomsian.com/blog/spring-boot-email-template>