

Instituto Tecnológico de Costa Rica
Campus Central Cartago
IC-7841: Proyecto de Ingeniería de Software



Manual Técnico - CliniPets Manager

William Alfaro Quirós – Carné: 2022437996

Jesus Cordero Díaz - Carné: 2020081049

Manfred Jones Sanabria - Carné: 2022176476

Kevelyn González Paizano - Carné: 2021034372

Andrey Rojas Garcia- Carné: 2021054965

Docente:

Alicia Salazar Hernández

Fecha de entrega:

30/01/2025

Verano, 2024-2025

1. Índice

1. Índice.....	2
2. Resumen.....	4
3. Descripción General del Sistema.....	5
3.1 Visión General.....	5
3.2 Arquitectura del Sistema.....	5
3.2.1 Modelo General de la Arquitectura.....	5
3.2.2 Tecnologías Utilizadas.....	5
3.3 Diagrama General de la Arquitectura del Software.....	6
4. Instalación y Configuración.....	7
4.1 Requisitos del Sistema.....	7
4.1.1 Hardware.....	7
4.1.2 Software.....	7
4.2 Pasos de Instalación.....	7
4.2.1 Instalación del Entorno de Desarrollo.....	7
4.3 Configuración de Variables de Entorno.....	8
4.4 Configuración de Firebase.....	9
5. Estructura del Código.....	10
5.1 Organización de Carpetas y Archivos.....	10
5.2 Módulos del Sistema.....	11
5.2.1 Gestión de Usuarios.....	11
5.2.2 Gestión de Citas.....	12
5.2.3 Gestión de Expedientes.....	12
5.2.4 Hojas de Internamiento.....	12
5.2.5 Informes Médicos.....	12
5.2.6 Gestión de Seguimiento de Vacunas.....	12
5.2.7 Inventario y Proveedores.....	13
5.2.8 Facturación.....	13
5.2.9 Gestión de Mascotas.....	13
5.2.10 Gestión de Notas de Seguimiento.....	13
5.2.11 Gestión de Especialistas y Salas.....	13
5.3 Explicación de los Principales Scripts y Funciones.....	14
6. Seguridad del Sistema.....	15
6.1 Control de Acceso y Roles de Usuario.....	15
6.2 Mecanismos de Autenticación y Autorización.....	15
6.3 Encriptación y Almacenamiento Seguro de Datos.....	16
7. Integraciones y Conectividad.....	17
7.1 Servicios Externos Utilizados.....	17
7.2 Métodos de Conexión y Comunicación Interna.....	17
8. Mantenimiento y Actualización.....	18
9. Pruebas y Validación.....	19
10. Anexos.....	20

10.1 Especificación de Requisitos de Software (ERS).....	20
10.2 Documento de Arquitectura del Sistema (SAD).....	20
10.3 Plan de Pruebas - CliniPets.....	20
10.4 Glosario de Términos Técnicos.....	20

2. Resumen

Este documento presenta el Manual Técnico del sistema CliniPets Manager, desarrollado como parte del curso IC-7841: Proyecto de Ingeniería de Software en el Instituto Tecnológico de Costa Rica. Su propósito es proporcionar a desarrolladores, administradores del sistema y personal técnico la información necesaria para comprender la arquitectura, instalación, mantenimiento y seguridad del sistema.

CliniPets Manager es un sistema diseñado para digitalizar y optimizar la gestión veterinaria, permitiendo administrar expedientes médicos, citas, inventario, proveedores, usuarios y el seguimiento de vacunación de las mascotas. Este manual es un recurso clave dentro del ciclo de vida del software, ya que facilita la implementación, el mantenimiento y futuras actualizaciones del sistema.

El documento se relaciona con otras entregas del proyecto, como la Especificación de Requisitos de Software (ERS), el Documento de Arquitectura del Sistema (SAD) y el Plan de Pruebas, garantizando que todos los aspectos del desarrollo sean documentados de manera estructurada.

En términos de alcance, este manual cubre aspectos técnicos como la instalación, configuración, estructura del código, seguridad, integración con Firebase, mantenimiento y pruebas del sistema. No incluye detalles sobre la interfaz de usuario o manuales de usuario, ya que su enfoque es completamente técnico.

3. Descripción General del Sistema

3.1 Visión General

CliniPets Manager es un sistema de gestión veterinaria diseñado para optimizar la administración de citas, expedientes médicos, inventario, proveedores, usuarios y el seguimiento de vacunación de mascotas. La implementación de este sistema permite mejorar la eficiencia operativa de la clínica, reducir errores manuales y asegurar un acceso rápido y seguro a la información.

El sistema se desarrolló utilizando metodologías **ágiles**, permitiendo iteraciones continuas para mejorar su rendimiento y funcionalidad. Su diseño modular facilita futuras ampliaciones y mantenimiento del software.

3.2 Arquitectura del Sistema

CliniPets Manager sigue una arquitectura en **capas**, basada en el patrón Modelo-Vista-Controlador (MVC), utilizando tecnologías modernas para garantizar escalabilidad, seguridad y facilidad de mantenimiento.

3.2.1 Modelo General de la Arquitectura

El sistema está dividido en tres capas principales:

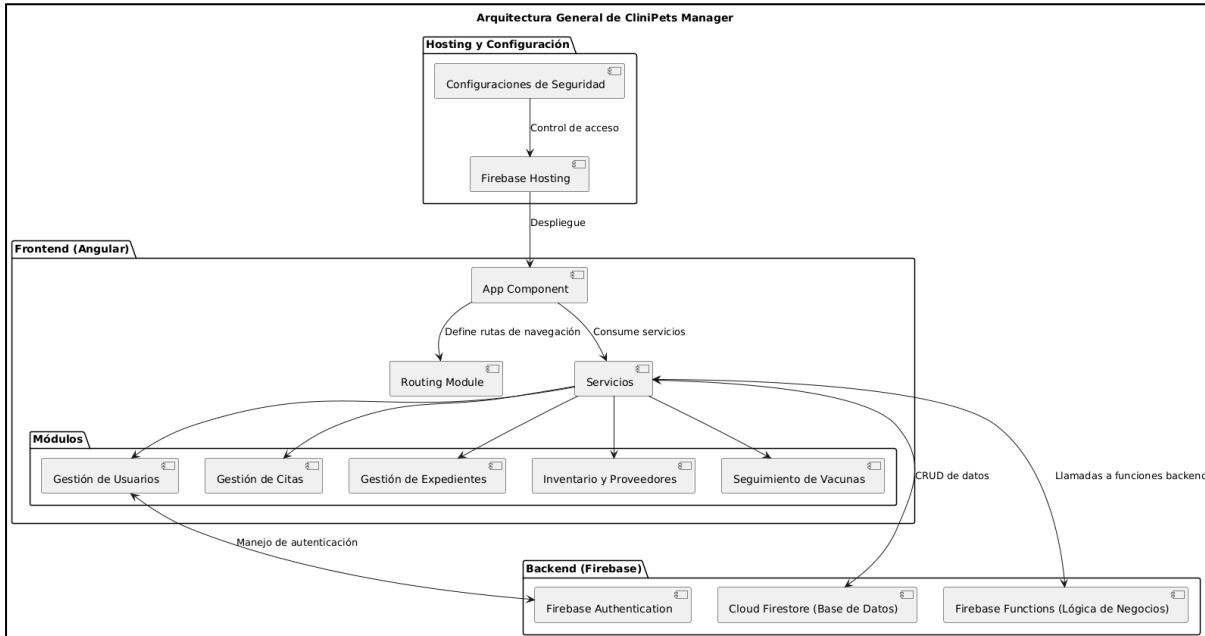
1. Capa de Presentación (Frontend): Desarrollada en Angular, encargada de la interfaz gráfica del usuario y la interacción con el sistema.
2. Capa de Lógica de Negocios (Backend): Se gestiona con Firebase Functions, proporcionando servicios para la autenticación, control de acceso y operaciones con la base de datos.
3. Capa de Datos: Utiliza Cloud Firestore de Firebase, una base de datos NoSQL en la nube que permite almacenamiento y acceso eficiente a la información.

3.2.2 Tecnologías Utilizadas

1. Frontend: Angular, TypeScript, Bootstrap.
2. Backend: Firebase Functions, Firebase Authentication.
3. Base de Datos: Firestore (NoSQL).
4. Control de Versiones: GitHub.

5. Hosting: Firebase Hosting.
6. Autenticación y Seguridad: Firebase Authentication con control de roles basado en permisos.

3.3 Diagrama General de la Arquitectura del Software



4. Instalación y Configuración

Esta sección detalla los requisitos y pasos necesarios para instalar y configurar CliniPets Manager, tanto en un entorno de desarrollo como en un entorno de producción.

4.1 Requisitos del Sistema

4.1.1 Hardware

1. Procesador: Intel Core i5 o superior.
2. RAM: Mínimo 8 GB (Recomendado 16 GB).
3. Almacenamiento: Al menos 10 GB de espacio libre.
4. Conexión a Internet estable.

4.1.2 Software

1. Sistema Operativo: Windows 10.
2. Node.js: Versión 22.12.0-64.msi.
3. Angular CLI: Version 19.0.5.
4. Firebase CLI: Version 11.1.0.
5. Git: Para control de versiones.
6. Visual Studio Code: Editor de código.

4.2 Pasos de Instalación

4.2.1 Instalación del Entorno de Desarrollo

1. Clonar el repositorio del proyecto.

```
git clone https://github.com/RojasAG/proyecto_ing.git  
  
cd proyecto_ing
```

2. Instalar Node.js y Angular CLI.

Si aún no tienes instalado Node.js, descárgalo desde nodejs.org e instálalo.

3. Instalar dependencias del proyecto.

Ejecuta el siguiente comando dentro de la carpeta del proyecto:

```
npm install --force
```

4. Construir el proyecto con Electrón:

Ejecute el siguiente comando para construir la carpeta "dist":

```
npm run electron
```

5. Comando para construir la aplicación de escritorio.

Ejecute el siguiente comando para construir la aplicación de escritorio:

```
npm run dist
```

4.3 Configuración de Variables de Entorno

En caso de querer cambiar la cuenta de Firebase a usar, asegúrate de actualizar los archivos de entorno en `src/environments/environment.ts`:

```
export const environment = {  
  production: false,  
  firebase: {  
    apiKey: "TU_API_KEY",  
    authDomain: "TU_PROYECTO.firebaseio.com",  
    projectId: "TU_PROYECTO",  
    storageBucket: "TU_PROYECTO.appspot.com",
```



```
messagingSenderId: "TU_MESSAGING_ID",  
  
appId: "TU_APP_ID"  
  
}  
  
};
```

4.4 Configuración de Firebase

1. Tener una cuenta de gmail vigente.
2. Tener la cuenta de google iniciada en el navegador.
3. Dirigirse a www.firebase.com.
4. Seleccionar la opción de "Go to console".
5. Crear un nuevo proyecto de firebase.
6. Después de crear el proyecto, seleccione "Agregar app".
7. En las opciones que se dan, seleccione la de tipo web.
8. Luego ingrese un nombre a la app a crearse.
9. Y listo, ya puede copiar y pegar los datos para realizar la conexión con firebase.

5. Estructura del Código

5.1 Organización de Carpetas y Archivos

El proyecto CliniPets Manager está estructurado en una arquitectura modular basada en Angular para el frontend y Firebase para el backend. A continuación, se describe la organización de las carpetas principales:

```
CliniPets-Manager/
├── .editorconfig
├── .firebaserc
├── .gitattributes
├── .gitignore
├── angular.json
├── firebase.json
├── package-lock.json
├── package.json
├── README.md
├── tsconfig.json
├── tsconfig.app.json
├── tsconfig.spec.json
├── functions/ # Funciones de Firebase para la lógica del backend
│   ├── index.js
│   ├── package.json
│   ├── package-lock.json
│   └── .eslintrc.js
├── public/ # Archivos estáticos
│   ├── clinicVet.jpg
│   ├── clinipets.png
│   ├── cliniPrueba.jpg
│   ├── favicon.ico
│   └── PerroGato.jpg
├── src/
│   ├── index.html
│   ├── main.server.ts
│   ├── main.ts
│   ├── server.ts
│   ├── styles.css
│   ├── environments/ # Configuración de entornos
│   │   ├── environment.prod.ts
│   │   └── environment.ts
│   ├── assets/ # Recursos adicionales
│   │   └── cliniPrueba.jpg
│   └── app/ # Carpeta principal del frontend
│       ├── app-routing.module.ts
│       ├── app.component.ts
│       └── app.module.ts
```

- admin-usuarios/
- citas/
- expedientes/
- hojas-internamiento/
- informes-medicos/
- seguimiento-vacunas/
- inventario/
- proveedores/
- facturas/
- lotes/
- mascotas/
- notas-seguimiento/
- productos/
- razas-mascotas/
- salas/
- especialistas/
- menu-principal/
- login/
- register-user/
- reset-password/
- models/ # Modelos de datos
- services/ # Servicios para manejo de datos
- shared/modal/ # Componentes compartidos

- interface/ # Definición de interfaces de datos

- appointment.interface.ts
- register.interface.ts
- reminder.interface.ts
- room.interface.ts
- specialist.interface.ts

5.2 Módulos del Sistema

El sistema está dividido en diferentes módulos según su funcionalidad. Cada módulo contiene su propio conjunto de componentes, servicios y modelos para mantener una estructura modular y escalable.

5.2.1 Gestión de Usuarios

1. Ubicación: src/app/admin-usuarios/
2. Funcionalidad: Control de usuarios y gestión de autenticación.
3. Archivos clave:
 - 3.1. admin-usuarios.component.ts (Componente principal)
 - 3.2. authlogin.service.ts (Servicio de autenticación)

5.2.2 Gestión de Citas

1. Ubicación: `src/app/citas/`
2. Funcionalidad: Registro, modificación y cancelación de citas médicas.
3. Archivos clave:
 - 3.1. `citas.component.ts` (Interfaz de citas)
 - 3.2. `citas.service.ts` (Manejo de datos de citas)

5.2.3 Gestión de Expedientes

1. Ubicación: `src/app/expedientes/`
2. Funcionalidad: Manejo de historiales médicos y notas de seguimiento.
3. Archivos clave:
 - 3.1. `expedientes.component.ts` (Interfaz de expedientes)
 - 3.2. `expedientes.service.ts` (Gestión de expedientes)

5.2.4 Hojas de Internamiento

1. Ubicación: `src/app/hojas-internamiento/`
2. Funcionalidad: Registro y seguimiento de internamientos.
3. Archivos clave:
 - 3.1. `hojas-internamiento.component.ts`
 - 3.2. `hojas-internamiento.service.ts`

5.2.5 Informes Médicos

1. Ubicación: `src/app/informes-medicos/`
2. Funcionalidad: Generación y consulta de informes médicos.
3. Archivos clave:
 - 3.1. `informes-medicos.component.ts`
 - 3.2. `informes-medicos.service.ts`

5.2.6 Gestión de Seguimiento de Vacunas

1. Ubicación: `src/app/seguimiento-vacunas/`
2. Funcionalidad: Control de vacunas aplicadas y futuras.
3. Archivos clave:
 - 3.1. `seguimiento-vacunas.component.ts`
 - 3.2. `seguimiento-vacunas.service.ts`

5.2.7 Inventario y Proveedores

1. Ubicación: src/app/inventario/, src/app/proveedores/
2. Funcionalidad: Control de stock, productos y gestión de proveedores.
3. Archivos clave:
 - 3.1. inventario.component.ts
 - 3.2. proveedores.component.ts
 - 3.3. productos.service.ts
 - 3.4. proveedores.service.ts

5.2.8 Facturación

1. Ubicación: src/app/facturas/
2. Funcionalidad: Registro y consulta de facturas generadas.
3. Archivos clave:
 - 3.1. facturas.component.ts
 - 3.2. factura.service.ts

5.2.9 Gestión de Mascotas

1. Ubicación: src/app/mascotas/
2. Funcionalidad: Registro y administración de mascotas.
3. Archivos clave:
 - 3.1. mascotas.component.ts
 - 3.2. mascotas.service.ts

5.2.10 Gestión de Notas de Seguimiento

1. Ubicación: src/app/notas-seguimiento/
2. Funcionalidad: Registro y consulta de notas médicas.
3. Archivos clave:
 - 3.1. notas-seguimiento.component.ts
 - 3.2. notas-seguimiento.service.ts

5.2.11 Gestión de Especialistas y Salas

1. Ubicación: src/app/especialistas/, src/app/salas/
2. Funcionalidad: Gestión del personal veterinario y salas disponibles.
3. Archivos clave:
 - 3.1. especialistas.component.ts

- 3.2. salas.component.ts
- 3.3. especialista.service.ts
- 3.4. salas.service.ts

5.3 Explicación de los Principales Scripts y Funciones

Además de los módulos específicos, el sistema incluye archivos esenciales como:

- 1. main.ts (Punto de entrada de Angular)
- 2. app.module.ts (Módulo principal)
- 3. functions/index.js (Lógica del backend con Firebase)

6. Seguridad del Sistema

El sistema CliniPets Manager implementa diversas medidas de seguridad para garantizar la protección de los datos, el acceso controlado y la integridad de la información. La seguridad se gestiona en varios niveles: autenticación, control de acceso y encriptación de datos.

6.1 Control de Acceso y Roles de Usuario

Para asegurar que cada usuario tenga acceso solo a la información y funcionalidades correspondientes a su rol, se implementa un sistema de control de acceso basado en roles (RBAC - Role-Based Access Control) utilizando Firebase Authentication.

Roles Definidos en el Sistema

Rol	Permisos
Administrador	Acceso total a todos los módulos, gestión de usuarios, configuración del sistema.
Asistente	Gestión de citas, expedientes, hojas de internamiento, informes médicos, seguimiento de vacunas, inventario, proveedores, facturas, lotes, mascotas, notas de seguimiento, productos, razas de mascotas y salas.
Ayudante	Gestión de citas, expedientes, hojas de internamiento, informes médicos, seguimiento de vacunas, inventario, proveedores, facturas, lotes, mascotas, notas de seguimiento, productos, razas de mascotas y salas.

Mecanismo de Autorización

1. Firebase Authentication autentica a los usuarios al iniciar sesión.
2. El sistema verifica el rol asignado al usuario en Firestore.
3. Cada módulo tiene restricciones de acceso según el rol del usuario.

6.2 Mecanismos de Autenticación y Autorización

El sistema implementa autenticación basada en Firebase Authentication, asegurando una gestión segura de credenciales y sesiones.

Autenticación

1. Se utiliza correo electrónico y contraseña como método principal de autenticación.
2. Se permite la recuperación de contraseña mediante un enlace enviado al correo del usuario.

Autorización

1. El acceso a los módulos está restringido según los permisos del usuario.
2. Los permisos se asignan en la base de datos Cloud Firestore.
3. La autorización se valida en el frontend antes de cargar cada módulo y en el backend antes de procesar solicitudes sensibles.

6.3 Encriptación y Almacenamiento Seguro de Datos

Encriptación de Datos Sensibles

1. Contraseñas: Firebase Authentication almacena y cifra automáticamente las credenciales.

7. Integraciones y Conectividad

El sistema CliniPets Manager es una aplicación independiente, diseñada exclusivamente para la gestión interna de una clínica veterinaria sin depender de servicios externos o integraciones con terceros. A continuación, se detallan los aspectos clave de su arquitectura de conectividad.

7.1 Servicios Externos Utilizados

Aunque el sistema no está diseñado para integraciones externas, hace uso de algunos servicios para su funcionalidad básica:

Servicio	Uso en el sistema
Firebase Authentication	Manejo de autenticación de usuarios y control de acceso basado en roles.
Firestore (Firebase Database)	Base de datos NoSQL donde se almacenan todos los datos del sistema.

Estos servicios se configuran internamente en Firebase y no requieren conexión con APIs o bases de datos externas.

7.2 Métodos de Conexión y Comunicación Interna

Dado que **CliniPets Manager** es una **Single Page Application (SPA) en Angular**, la comunicación entre el frontend y la base de datos Firestore se realiza de manera directa a través de los **Servicios de Firebase**.

Flujo de comunicación en el sistema:

1. **El usuario accede a la aplicación** y Firebase Authentication gestiona la autenticación.
2. **El frontend (Angular)** envía peticiones directamente a Firestore a través de los servicios de Angular.
3. **Firestore almacena y recupera los datos** en tiempo real según las reglas de acceso definidas.

Este flujo garantiza **baja latencia**, ya que Firebase permite la comunicación **en tiempo real** con la base de datos.

8. Mantenimiento y Actualización

El sistema CliniPets Manager ha sido desarrollado como un proyecto académico en un tiempo limitado, por lo que no se contempla realizar actualizaciones o mantenimiento una vez entregado el código.

9. Pruebas y Validación

El proceso de pruebas y validación en CliniPets Manager se llevó a cabo siguiendo una estrategia de pruebas de aceptación, con el objetivo de garantizar que el sistema cumpla con los requisitos establecidos en la especificación de requisitos (ERS).

En caso de querer profundizar más con relación con las pruebas y validaciones ver el documento Plan de Pruebas - CliniPets

10. Anexos

Para obtener información detallada sobre los requisitos del sistema, la arquitectura del software y las estrategias de prueba implementadas, se recomienda consultar los siguientes documentos:

10.1 Especificación de Requisitos de Software (ERS)

1. Contiene la definición de los requisitos funcionales y no funcionales del sistema.
2. Describe los módulos principales y su propósito dentro del sistema.
3. Presenta restricciones y características de los usuarios esperados.

10.2 Documento de Arquitectura del Sistema (SAD)

1. Detalla la estructura del sistema, incluyendo su arquitectura en capas.
2. Presenta diagramas UML sobre la organización de los módulos y su interacción.
3. Explica las decisiones arquitectónicas y su justificación técnica.

10.3 Plan de Pruebas - CliniPets

1. Contiene la estrategia de pruebas implementada en el proyecto.
2. Lista los casos de prueba aplicados junto con sus criterios de éxito.
3. Describe los procedimientos de validación y control de calidad.

10.4 Glosario de Términos Técnicos

A continuación, se presentan algunos términos clave utilizados en este documento y en el desarrollo de CliniPets Manager:

Término	Definición
Firebase Authentication	Servicio de autenticación utilizado para gestionar los accesos al sistema.
Firestore	Base de datos NoSQL utilizada para almacenar la información de CliniPets Manager.

SPA (Single Page Application)	Tipo de aplicación web que carga una sola vez y actualiza contenido dinámicamente.
RBAC (Role-Based Access Control)	Modelo de control de acceso basado en roles definidos para cada usuario.
Firestore Rules	Reglas de seguridad que definen qué usuarios pueden leer o modificar datos en Firebase.