



**FACULTAD DE
INGENIERÍA**

UNIVERSIDAD DA VINCI
DE GUATEMALA

Universidad Da Vinci De Guatemala

Facultad de Ingeniería

Carrera: Ingeniería de sistemas



**FACULTAD DE
INGENIERÍA**

UNIVERSIDAD DA VINCI
DE GUATEMALA

“Examen final ”

Nombre completo: Francis Mariana Zeceña Cifuentes

Número de carnet:202406016

Curso: Programación web

Guatemala, diciembre de 2025



Introducción

La presente propuesta desarrolla una solución web orientada a resolver una necesidad real dentro del contexto guatemalteco: la falta de un sistema centralizado y eficiente para gestionar procesos de adopción de animales. Actualmente, la mayoría de refugios y rescatistas utilizan redes sociales para publicar mascotas, lo que provoca desorden, pérdida de información y dificulta que las personas interesadas encuentren un animal adecuado para adoptar.

El proyecto *AdoptaGT* surge como una alternativa tecnológica que busca mejorar este proceso mediante una plataforma web moderna, accesible y fácil de usar. A través de esta herramienta, se pretende conectar de manera más efectiva a rescatistas con posibles adoptantes, ofreciendo un flujo claro de información y un proceso de adopción más responsable.



Objetivo

El proyecto tiene como objetivo evaluar mi capacidad para conceptualizar, planificar y justificar técnicamente una solución web profesional. La propuesta consiste en crear una aplicación web llamada *AdoptaGT*, cuyo propósito es facilitar la adopción responsable de animales en Guatemala. Esta plataforma permitirá que refugios y rescatistas publiquen animales disponibles, y que las personas interesadas puedan verlos, filtrarlos y enviar solicitudes de adopción de forma sencilla y organizada.



Problema

En Guatemala existen numerosos animales abandonados que dependen de rescatistas y refugios para encontrar un hogar. Sin embargo, el proceso actual de adopción es desordenado porque la mayoría de publicaciones se realizan en redes sociales, donde la información se pierde con facilidad. Esto provoca que las personas interesadas tengan dificultades para encontrar mascotas disponibles, que no existan filtros claros para buscar opciones y que no se tenga un control adecuado sobre las solicitudes de adopción.

Solución

La propuesta consiste en desarrollar una aplicación web llamada *AdoptaGT*, un sistema centralizado donde los refugios y rescatistas podrán registrar mascotas disponibles y los usuarios podrán verlas, filtrarlas y enviar solicitudes de adopción. La plataforma permitirá organizar el proceso, mantener un historial de cada caso y facilitar una comunicación más clara entre adoptantes y rescatistas.

Justificación

AdoptaGT aporta valor porque ofrece una solución práctica a un problema social real. El sistema mejora la visibilidad de los animales, agiliza el proceso de adopción y permite una gestión más responsable. Además, al centralizar la información en un solo lugar, el proyecto beneficia tanto a las organizaciones de rescate como a las personas que desean adoptar de forma informada y segura.



Arquitectura Tecnológica

La aplicación propuesta utilizará una arquitectura moderna que separa claramente el frontend, el backend y la base de datos. Esta estructura facilita el mantenimiento, la escalabilidad y el desarrollo de nuevas funciones en el futuro. La elección de tecnologías se basa en su estabilidad, facilidad de uso y disponibilidad de documentación.

Frontend

Para el frontend se utilizará React junto con Vite. Esta combinación permite crear interfaces rápidas, modernas y fáciles de mantener. React ofrece componentes reutilizables que facilitan la construcción de páginas como la lista de mascotas, los perfiles individuales y los formularios de adopción. Vite acelera el desarrollo y mejora el rendimiento general de la aplicación.

Backend

El backend será desarrollado con Node.js y Express. Estas tecnologías permiten crear una API que se comunica de manera eficiente con el frontend. Express es ligero, flexible y ampliamente utilizado, lo que lo hace adecuado para manejar solicitudes, gestionar rutas y procesar datos enviados por los usuarios o refugios.

Base de Datos

Para la gestión de los datos se utilizará PostgreSQL, una base de datos relacional confiable y robusta. Esta opción es adecuada porque el proyecto maneja datos estructurados que requieren relaciones claras, como usuarios, mascotas, refugios y solicitudes de adopción. PostgreSQL facilita consultas complejas y asegura la integridad de la información almacenada.

Diseño Técnico

El diseño técnico del proyecto establece cómo se organizará la información dentro del sistema y cómo se comunicarán los distintos componentes de la aplicación. Esta sección describe el modelo de base de datos y los endpoints principales que forman la API de la plataforma. Ambos elementos son esenciales para que la aplicación funcione de manera ordenada y permita administrar correctamente el proceso de adopción.

Modelo de Base de Datos



La base de datos utiliza un modelo relacional que permite mantener relaciones claras entre usuarios, refugios, mascotas y solicitudes de adopción. Este modelo ayuda a garantizar la integridad de los datos y facilita consultas como identificar mascotas disponibles, solicitudes pendientes o adopciones completadas.

La tabla de usuarios almacena la información de quienes interactúan con el sistema, ya sean adoptantes, administradores o representantes de refugios. La tabla de refugios registra los datos de las organizaciones o personas que publican animales. La tabla de mascotas contiene la información principal de cada animal, como nombre, especie, tamaño, descripción y estado. Finalmente, la tabla de solicitudes registra cada vez que un usuario aplica para adoptar una mascota.

A continuación se presentan ejemplos de cómo pueden verse algunos registros dentro de la tabla de mascotas utilizando nombres reales de animales:

```
{
  "id_mascota": 1,
  "nombre": "Tutis",
  "especie": "gato",
  "edad_aproximada": 2,
  "tamaño": "pequeño",
  "estado": "disponible"
}
```

```
{
  "id_mascota": 2,
  "nombre": "Sergia",
  "especie": "gato",
  "edad_aproximada": 1,
  "tamaño": "pequeño",
  "estado": "disponible"
}
```



```
{"id_mascota": 3, "nombre": "Pollito Dexter", "especie": "gato", "edad_aproximada": 3, "tamaño": "mediano", "estado": "en_proceso"}
```

```
{"id_mascota": 4, "nombre": "Tomsito", "especie": "gato", "edad_aproximada": 1, "tamaño": "pequeño", "estado": "disponible"}
```

```
{"id_mascota": 5, "nombre": "Cocoa", "especie": "gato", "edad_aproximada": 2, "tamaño": "mediano", "estado": "disponible"}
```

```
{"id_mascota": 6, "nombre": "Jorgito", "especie": "gato", "edad_aproximada": 5, "tamaño": "grande", "estado": "adoptado"}
```

Este tipo de registros permite mostrar ejemplos reales dentro del sistema, demostrando cómo se vería la información cuando la plataforma esté en funcionamiento.

Diseño de la API

La API es la capa encargada de permitir la comunicación entre el frontend desarrollado en React y el backend creado con Node.js y Express. A través de esta API se envían y reciben datos como el listado de mascotas, las solicitudes de adopción y los cambios de estado de los animales.

A continuación se describen tres endpoints fundamentales:

Endpoint 1: Obtener lista de mascotas disponibles

Método: GET

Ruta: /api/v1/mascotas

Descripción: Devuelve las mascotas que están disponibles para adopción. Permite aplicar filtros como especie, tamaño o ubicación.

Ejemplo de respuesta:

```
{"id_mascota": 5, "nombre": "Cocoa", "especie": "gato", "edad_aproximada": 2, "tamaño": "mediano", "estado": "disponible"}
```

Endpoint 2: Crear una solicitud de adopción

Método: POST

Ruta: /api/v1/solicitudes

Descripción: Registra una solicitud enviada por un usuario que desea adoptar una mascota.

Ejemplo de cuerpo enviado por el usuario:



```
{"id_mascota": 4, "id_usuario": 10, "mensaje": "Deseo aplicar para adoptar a Tomsito. Tengo experiencia cuidando gatos y un hogar adecuado."}
```

Ejemplo de respuesta del servidor:

```
{"id_solicitud": 12, "estado_solicitud": "pendiente"}
```

Endpoint 3: Actualizar el estado de una mascota

Método: PATCH

Ruta: /api/v1/mascotas/:id_mascota/estado

Descripción: Permite modificar el estado de una mascota, por ejemplo cambiar de disponible a adoptado.

Ejemplo del cuerpo enviado:

```
{"estado": "adoptado"}
```

Ejemplo de respuesta:

```
{"mensaje": "El estado de Pollito Dexter se actualizó correctamente."}
```



Gerencia del Proyecto (Horas de desarrollo y Costos)

Gerencia del Proyecto

La gerencia del proyecto permite estimar el tiempo necesario para desarrollar la plataforma y calcular un costo aproximado basado en las horas de trabajo requeridas. Esta planificación ayuda a comprender el esfuerzo total involucrado en la creación del sistema, desde la etapa inicial de análisis hasta la implementación final.

Estimación de horas de desarrollo

El desarrollo del proyecto se divide en diferentes fases. Cada fase representa una actividad esencial para lograr una versión funcional de la plataforma. La estimación se basa en el tiempo promedio que toma realizar cada tarea utilizando tecnologías como React para el frontend, Node.js y Express para el backend, y PostgreSQL para la base de datos.

Las horas estimadas son las siguientes:

Análisis del problema y definición de la solución: 6 horas

Diseño de la arquitectura tecnológica y el modelo de datos: 8 horas

Diseño y documentación de la API: 6 horas

Desarrollo del frontend en React: 18 horas

Desarrollo del backend con Node.js y Express: 18 horas

Integración entre frontend y backend: 10 horas

Pruebas generales de funcionamiento: 6 horas

Despliegue de la aplicación y configuración del entorno: 4 horas

Documentación técnica y preparación del material de presentación: 6 horas

El total estimado es de 78 horas de trabajo.



Estimación de costos

Para calcular el costo del proyecto se utiliza una tarifa de Q50.00 por hora, que representa un valor promedio para un desarrollador independiente en un proyecto de este tipo. El costo base se obtiene multiplicando la tarifa por la cantidad total de horas.

Costo base:

$$78 \text{ horas} \times \text{Q}50.00 = \text{Q}3,900.00$$

Para cubrir imprevistos, ajustes adicionales o tareas no contempladas, se agrega un 20% de colchón.

Cálculo del colchón:

$$20\% \text{ de } \text{Q}3,900.00 = \text{Q}780.00$$

Costo total estimado del proyecto:

$$\text{Q}3,900.00 + \text{Q}780.00 = \text{Q}4,680.00$$

Este valor representa el costo aproximado de desarrollar la primera versión funcional de la plataforma AdoptaGT.



Conclusión

El proyecto *AdoptaGT* presenta una solución tecnológica clara y bien estructurada para mejorar el proceso de adopción de animales en Guatemala. A través de una arquitectura moderna, un modelo de datos organizado y una API funcional, la plataforma permite centralizar la información de mascotas, agilizar las solicitudes de adopción y fortalecer la comunicación entre refugios y adoptantes. Además, la planificación de horas y costos demuestra que el desarrollo es viable y puede implementarse de manera realista.

Esta propuesta establece las bases necesarias para crear una herramienta que aporte beneficios sociales y tecnológicos, al mismo tiempo que abre la posibilidad de futuras mejoras como integración de donaciones, seguimientos post-adopción y funciones adicionales para los refugios. El proyecto cumple con los requisitos establecidos y representa una solución innovadora con impacto positivo en la comunidad.