



**FACULTAD DE
INGENIERÍA**

UNIVERSIDAD DA VINCI
DE GUATEMALA

Universidad Da Vinci De Guatemala

Facultad de Ingeniería

Carrera: Ingeniería en sistemas.



**FACULTAD DE
INGENIERÍA**

UNIVERSIDAD DA VINCI
DE GUATEMALA

“Propuesta de Proyecto de Innovación Web

Proyecto: FinTrack Estudiantil – Control de Finanzas Personales”

Carlos Eduardo Toj Dardón

202403768

Curso: Programación Web

Guatemala, 07 de diciembre de 2025



Introducción

El presente documento tiene como objetivo presentar la propuesta técnica del proyecto FinTrack Estudiantil, una aplicación web diseñada para apoyar a los estudiantes en la gestión de sus finanzas personales. La iniciativa surge a partir de la necesidad común de contar con una herramienta sencilla y accesible que permita registrar gastos e ingresos, visualizar resúmenes mensuales y mantener un mejor control del presupuesto.

A lo largo del documento se describen los elementos fundamentales de la planificación del proyecto, iniciando con la definición del problema y la solución propuesta. Posteriormente, se detalla la arquitectura tecnológica seleccionada, incluyendo las decisiones relacionadas con el frontend, backend y la persistencia de datos. También se presenta el diseño técnico del sistema mediante el modelo de base de datos y la estructura de la API. Finalmente, se realiza una estimación del esfuerzo y del costo necesario para desarrollar el prototipo inicial.

Esta propuesta sirve como base para el desarrollo del MVP, permitiendo visualizar de manera clara el alcance del proyecto y las decisiones técnicas que guiarán su implementación. Asimismo, representa un ejercicio práctico de planificación y diseño en el contexto del desarrollo web moderno.



Pitch del Proyecto (Problema, Solución y Justificación)

El Problema

En la universidad es muy común que los estudiantes lleven sus finanzas “a la ligera”: gastan en comida rápida, transporte, suscripciones y pequeños antojos sin tener un registro claro. La mayoría solo revisa el estado de cuenta del banco cuando ya está casi en cero o cuando aparece un cobro que no esperaban.

Esta falta de control provoca que se queden sin dinero antes de fin de mes, no sepan en qué se les fue el sueldo o la mesada y les cueste separar dinero para metas importantes como comprar equipo, pagar estudios o ahorrar para emergencias.

La Solución

La solución propuesta es FinTrack Estudiantil, una aplicación web sencilla donde el usuario puede registrar sus ingresos y gastos día a día, clasificarlos por categoría (comida, transporte, entretenimiento, estudio, etc.) y visualizar resúmenes mensuales.

La aplicación mostrará un balance actualizado, gráficos básicos de cómo se distribuyen los gastos y alertas simples cuando el usuario se acerque al límite de presupuesto que definió para el mes.

Justificación e Innovación

FinTrack Estudiantil busca ser una herramienta enfocada específicamente en estudiantes y jóvenes que manejan montos relativamente pequeños, pero muy variables. A diferencia de una hoja de cálculo genérica, la aplicación permite registrar movimientos de forma rápida desde el navegador, ver estadísticas visuales y configurar presupuestos por categoría de una manera más amigable. La idea es innovadora en mi contexto porque combina buenas prácticas de desarrollo web (frontend, backend y base de datos) con una necesidad real que vivimos a diario: aprender a administrar nuestro dinero desde temprano. Además, sirve como base para un futuro proyecto más grande donde se podrían integrar metas de ahorro, recordatorios de pagos y conexión con datos bancarios.



Arquitectura Tecnológica

Frontend

Para el desarrollo del frontend se evaluaron tres opciones populares en el ecosistema web moderno: React, Vue y Angular.

- React es una biblioteca muy utilizada para construir interfaces de usuario basadas en componentes. Tiene una comunidad enorme, abundante documentación y se integra fácilmente con otras librerías del ecosistema JavaScript.
- Vue es un framework progresivo, sencillo de aprender y con una curva de aprendizaje amigable, pero en el mercado laboral local no es tan demandado como React.
- Angular es un framework completo y robusto, pero su estructura es más pesada y puede resultar excesiva para un proyecto tipo MVP como el que se plantea.

Elección final:

Se utilizará React para el frontend de FinTrack Estudiantil, porque permite construir una SPA (Single Page Application) modular y reactiva, con un ecosistema maduro y mucho material disponible para resolver problemas rápidamente. Además, ya tengo experiencia previa trabajando con React, lo que reduce el riesgo y el tiempo de desarrollo.

Backend y Publicación

Para el backend se utilizará Node.js con el framework Express. Esta combinación facilita la creación de APIs RESTful que se comunican usando JSON, lo cual encaja bien con el frontend en React. Node.js es ligero, tiene un único lenguaje (JavaScript) en todo el stack y posee gran soporte de paquetes mediante npm.

Para la publicación del proyecto se considera el despliegue en un servicio en la nube como Render o Vercel, que permiten hospedar tanto el frontend como el backend con configuraciones relativamente sencillas para proyectos estudiantiles. En una fase posterior también podría desplegarse en otras plataformas como AWS o Azure si se requiere mayor escalabilidad.



Persistencia de Datos

Para la persistencia de la información se optará por una base de datos relacional (SQL), utilizando motores como MySQL o PostgreSQL.

Los datos del sistema (usuarios, movimientos financieros, categorías, presupuestos) tienen una estructura clara y requieren relaciones definidas, por ejemplo: un usuario tiene muchos movimientos y cada movimiento pertenece a una categoría.

Usar SQL permite:

- Definir integridad referencial entre tablas.
- Realizar consultas con filtros por fecha, categoría o rango de montos.
- Obtener resúmenes mensuales de gastos e ingresos de forma eficiente.

Por estas razones, una base de datos SQL se ajusta mejor que un modelo NoSQL para el tipo de información y consultas que manejará FinTrack Estudiantil.



Diseño Técnico

Modelo de Base de Datos (ERD)

Descripción del Modelo de Datos

El modelo de datos se diseñó considerando que cada usuario tendrá sus propios movimientos financieros (ingresos y gastos), categorizados y organizados cronológicamente. Además, se incluyen presupuestos mensuales por categoría para permitir alertas y análisis simples.

Las entidades principales son:

Usuarios (users)

Contiene los datos de cada usuario registrado.

Campo	Tipo	Descripción
Id_user (PK)	INT	Identificador único del usuario
name	VARCHAR	Nombre del usuario
email	VARCHAR	Correo único
password	VARCHAR	Contraseña en hash

Categorías (categories)

Contiene los datos de cada usuario registrado.

Campo	Tipo	Descripción
id_category (PK)	INT	Identificador de categoría
name	VARCHAR	Nombre de categoría
type	ENUM("ingreso", "gasto")	Tipo de movimiento



Movimientos Financieros (movements)

Registra cada gasto o ingreso realizado por el usuario.

Campo	Tipo	Descripción
id_movement (PK)	INT	Identificador del movimiento
id_user (FK)	INT	Usuario al que pertenece
id_category (FK)	INT	Categoría del movimiento
amount	DECIMAL	Monto
description	VARCHAR	Descripción opcional
date	DATE	Fecha del movimiento

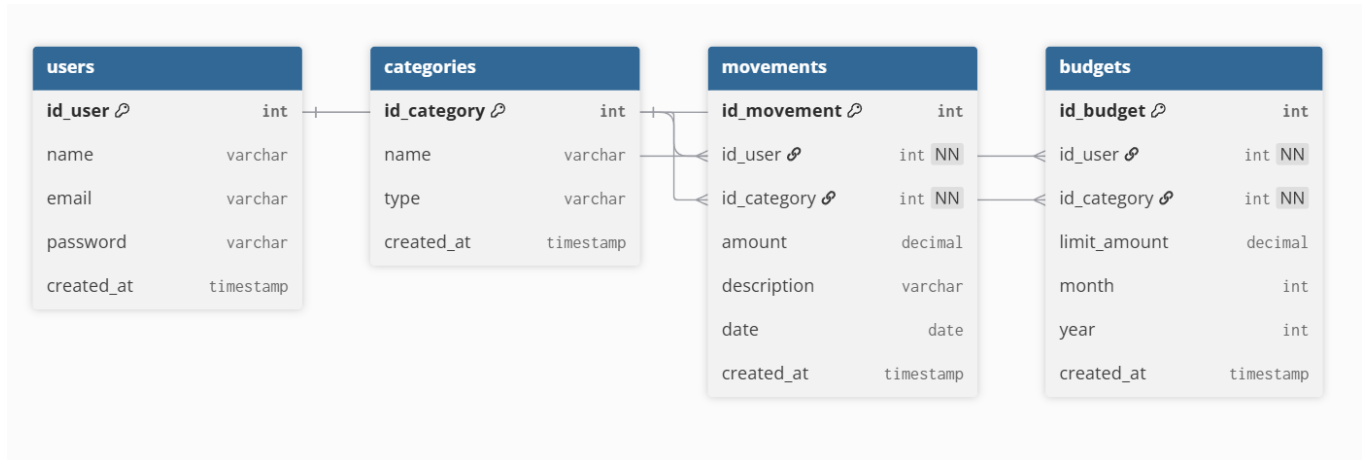
Presupuestos (budgets)

Para definir límites de gasto por categoría.

Campo	Tipo	Descripción
id_budget (PK)	INT	Identificador
id_user (FK)	INT	Usuario
id_category (FK)	INT	Categoría
limit_amount	DECIMAL	Monto máximo mensual permitido
month	INT	Mes (1–12)
year	INT	Año



Diagrama ERD



El modelo de datos de FinTrack Estudiantil está compuesto por cuatro entidades principales: usuarios, categorías, movimientos financieros y presupuestos. Cada usuario puede registrar múltiples movimientos y presupuestos, mientras que las categorías permiten clasificar los gastos e ingresos. El diagrama muestra las relaciones entre las tablas mediante llaves foráneas que garantizan consistencia referencial.



Diseño de API

Endpoint 1: Obtener movimientos del usuario

- Método: GET
- Ruta: /api/v1/movements
- Descripción: Retorna la lista de movimientos financieros (ingresos y gastos) del usuario autenticado. Opcionalmente se puede filtrar por mes y año.

Respuesta Exitosa (200 OK):

Formato JSON

```
[
  {
    "id_movement": 101,
    "category": "Comida",
    "type": "gasto",
    "amount": 35.50,
    "description": "Almuerzo en la universidad",
    "date": "2024-12-04"
  },
  {
    "id_movement": 102,
    "category": "Salario",
    "type": "ingreso",
    "amount": 1200.00,
    "description": "Pago quincenal",
    "date": "2024-12-03"
  }
]
```



Endpoint 2: Registrar un nuevo movimiento

- Método: POST
- Ruta: /api/v1/movements
- Descripción: Registra un nuevo ingreso o gasto asociado al usuario autenticado.

Cuerpo de la Petición (Body):

Formato JSON

```
{
  "id_category": 3,
  "amount": 45.00,
  "description": "Pago de transporte",
  "date": "2024-12-05"
}
```

Respuesta Exitosa (201 Created):

Formato JSON

```
{
  "message": "Movimiento registrado exitosamente",
  "movement_id": 103
}
```



Endpoint 3: Obtener resumen mensual

- Método: GET
- Ruta: /api/v1/summary
- Descripción: Devuelve un resumen mensual con el total de ingresos, el total de gastos y el balance del usuario para un mes y año específicos.

Respuesta Exitosa (200 OK):

Formato JSON

```
{  
  "month": 12,  
  "year": 2024,  
  "total_ingresos": 1500.00,  
  "total_gastos": 780.50,  
  "balance": 719.50  
}
```



Planificación y Costos

Estimación de Esfuerzo

A continuación, se presenta el desglose de tareas necesarias para el desarrollo del prototipo v0 de FinTrack Estudiantil. Las horas estimadas se basan en un nivel de complejidad junior y un ritmo realista de trabajo individual.

No.	Tarea	Descripción breve	Horas estimadas
1	Análisis del problema	Revisión del caso, definición del alcance y requerimientos.	3 horas
2	Diseño del modelo de datos	Construcción del ERD y validación de entidades y relaciones.	3 horas
3	Diseño de la API	Definición de endpoints, rutas, métodos y formatos JSON.	3 horas
4	Configuración inicial del proyecto	Creación del proyecto React, configuración de dependencias y estructura.	4 horas
5	Desarrollo del Frontend (v0)	Pantallas básicas: registro de movimientos y listado.	8 horas
6	Desarrollo del Backend (v0)	Implementación de los 3 endpoints principales.	8 horas
7	Integración Frontend-Backend	Conexión de API, pruebas funcionales básicas.	4 horas



FACULTAD DE INGENIERÍA

UNIVERSIDAD DA VINCI
DE GUATEMALA

8	Pruebas y corrección de errores	Ajustes menores, depuración de flujos principales.	3 horas
9	Despliegue del prototipo	Publicación de frontend y backend en plataforma cloud.	2 horas
10	Preparación del documento PDF	Redacción técnica, diagramas, limpieza visual.	3 horas
11	Creación de diapositivas	Diseño del pitch y resumen técnico.	2 horas
12	Grabación del video pitch	Presentación del problema, solución y demo.	2 horas

Total, inicial de horas estimadas:

45 horas

Aplicación de Colchón del 20%:

45 horas + 20% = 54 Total horas estimadas con colchón.

Presupuesto del Proyecto

Para calcular el presupuesto, se define un costo por hora del desarrollador.

Costo por hora:

Q75 por hora

Cálculo de presupuesto total:

54 horas x Q 75/hora = Q 4,050 ya con margen del 20%

Presupuesto Final del Proyecto

Q 4,050



Conclusión

La elaboración de esta propuesta me permitió comprender de forma más clara cómo un proyecto web debe planificarse desde una perspectiva técnica y organizacional. A lo largo del desarrollo del documento definí el problema, diseñé la arquitectura tecnológica, construí el modelo de datos y estructuré una API coherente con las necesidades de FinTrack Estudiantil. Este proceso me ayudó a visualizar mejor cómo se relacionan los componentes de un sistema real y a dimensionar el esfuerzo necesario para llevar un prototipo a una versión funcional.

Considero que la solución planteada es viable y aporta valor para los estudiantes que buscan administrar mejor sus finanzas personales. Además, este proyecto representa una base sólida para continuar aprendiendo y ampliarlo más adelante con nuevas funcionalidades. Con esta experiencia, reafirmo la importancia de planificar correctamente antes de desarrollar, ya que una buena arquitectura y un alcance bien definido facilitan mucho el trabajo futuro.