



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Universidad de las Fuerzas Armadas “ESPE”
Departamento de las Ciencias de la Computación



**Desarrollo Web
Avanzado**

Título del informe: Tablero de Control Financiero

Asignatura: Desarrollo Web Avanzado

Nombre del autor o autores:

- Alexis Chimba
- Ariel Jara

Fecha de entrega: 26/01/2025



2. Índice

1. Resumen.
2. Introducción
3. Objetivos
4. Desarrollo
5. Resultados
6. Conclusiones
7. Recomendaciones

3. Resumen

En el proyecto vamos a realizar un tablero que tenga un control financiero para que pueda realizar los ingresos, gastos y estadísticas financiera en la cual se utiliza componentes interactivos que son de **PrimeNG** en los cuales se puede utilizar gráficos (Chart), también se utilizan tarjetas (Card), también vamos aplicar campos numéricos (InputNumber), se va utilizar tablas para poder filtrar datos (DataTable) y botones (ToogleButton).

4. Introducción

PrimeNG mejora las aplicaciones web, en donde es conjunto completo donde contiene componentes en una interfaz de usuarios que son personalizados y son funciones de PrimeNG, convierte su visión de desarrollo a una realidad.





Angular proporciona con un amplio conjunto donde contiene herramientas, API donde se simplifica y se agiliza con un flujo de trabajo de desarrollo, angular es una plataforma sólida donde se crean aplicaciones rápidas.



5. Objetivos

- **General**

1. Se va a desarrollar un tablero financiero interactivo donde se van a mostrar las estadísticas de manera eficiente.
2. Se proporciona una herramienta intuitiva donde se facilita la gestión financiera.
3. Mejora la experiencia del usuario con una implementación de gráficos dinámicos y con tablas filtradas.
4. Se creará un sistema que se adapte y que sea escalable con diferentes necesidades financieras.

- **Específicos**

1. Se diseña con gráficos interactivos para que representen los ingresos con gastos en diferentes formas: líneas, barras.
2. Se implementaron tarjetas que sean dinámicas donde se mostraran métricas claves como el total de ingresos y gastos.
3. Se propone desarrollar campos de entrada donde se registrará las cantidades y datos que son relevantes en las transacciones.
4. Se implementa un botón que permita cambiar entre vistas detalladas, resumidas según lo que quiera mostrar el usuario.

6. Desarrollo



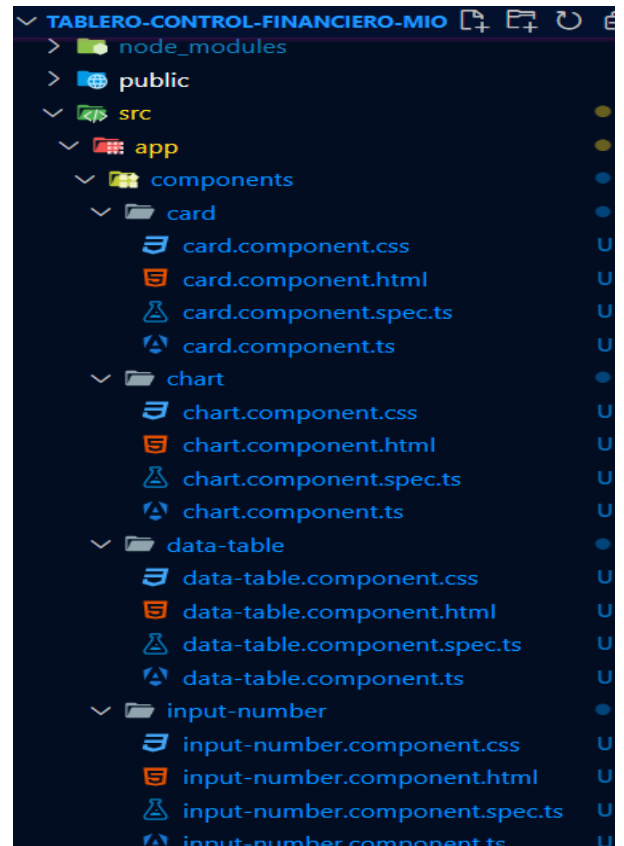
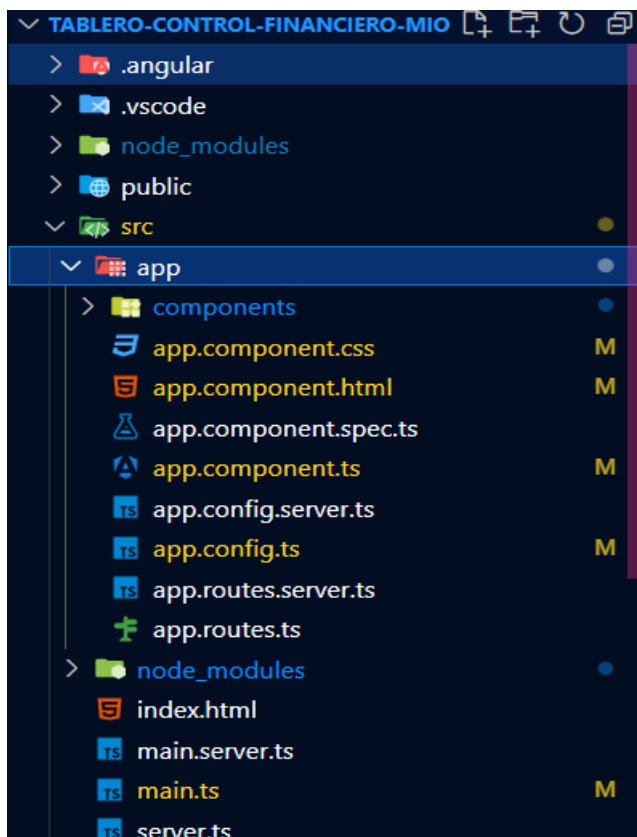
El Tablero de Control Financiero que es un sistema que se ha diseñado que proporciona una representación visual que es interactiva con los datos, donde permiten analizar ingresos y gastos con un gráfico interactivo con tablas y funcionalidades.

En el componente principal que es AppComponent se gestiona el tablero global donde se distribuye todas las responsabilidades con los componentes que son secundarios en los cuales tienen funciones específicas:

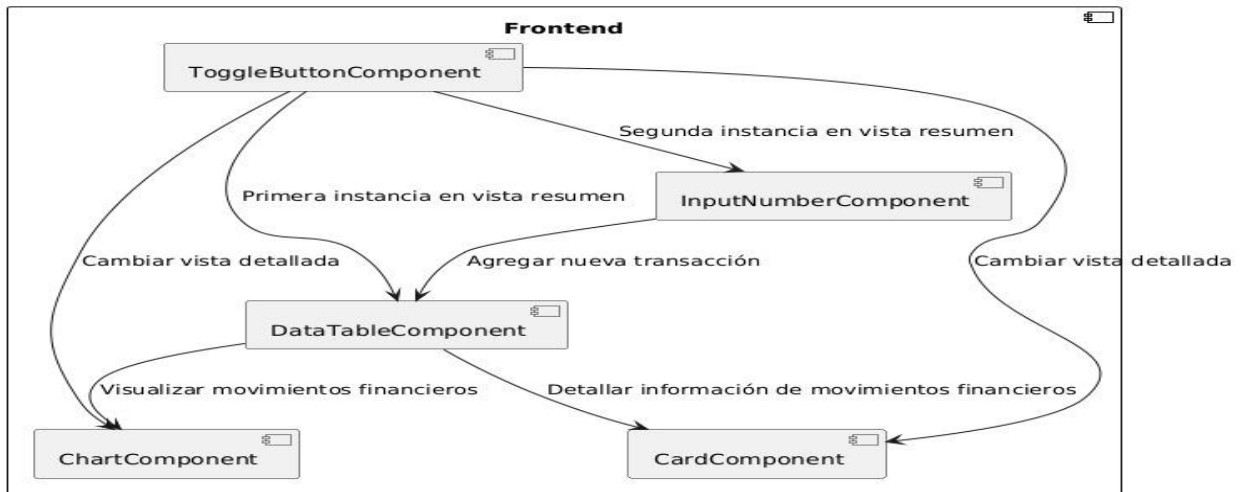
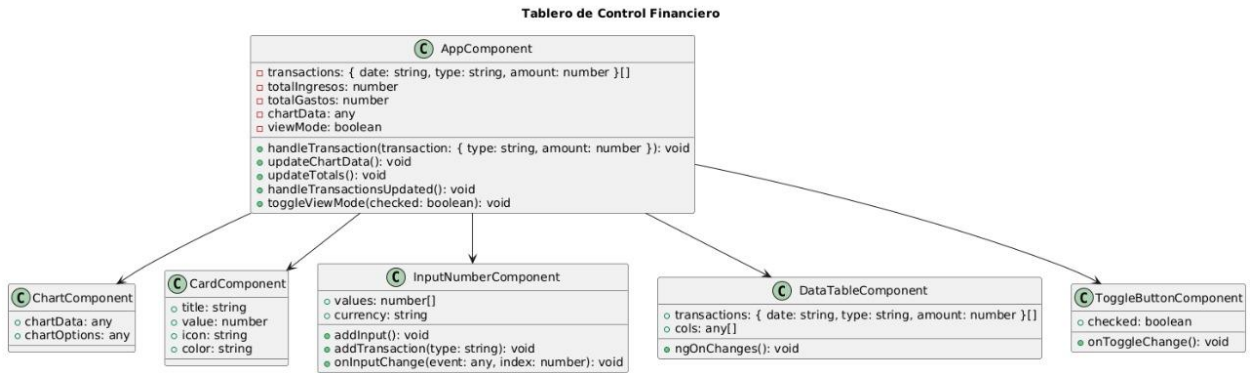
- AppComponent

Componentes:

- ChartComponent
- CardComponent
- InputNumberComponent
- DataTableComponent
- ToggleButtonComponent



Diagramas



Comandos para Instalar PrimeNG

- npm install primeng icons @angular/cdk
- npm install primeng icons @angular/cli
- npm install primeng@19.0.3



```
Browser bundles
Initial chunk files | Names | Raw size
polyfills.js | polyfills | 90.20 kB |
main.js | main | 44.25 kB |
styles.css | styles | 95 bytes |

| Initial total | 134.54 kB

Server bundles
Initial chunk files | Names | Raw size
polyfills.server.mjs | polyfills.server | 572.91 kB |
main.server.mjs | main.server | 45.79 kB |
server.mjs | server | 1.86 kB |
```

Codigo

App.component.ts

```
You, hace 4 horas | 2 authors (You and one other)
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { ChartComponent } from '../components/chart/chart.component';
import { CardComponent } from '../components/card/card.component';
import { InputNumberComponent } from '../components/input-number/input-number.component';
import { DataTableComponent } from '../components/data-table/data-table.component';
import { ToggleButtonComponent } from '../components/toggle-button/toggle-button.component';
import { CommonModule } from '@angular/common';

@Component({
  selector: 'app-root',
  imports: [RouterOutlet, ChartComponent, CardComponent, InputNumberComponent, DataTableComponent, ToggleButtonComponent],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  transactions: { date: string, type: string, amount: number }[] = [];
  totalIngresos: number = 0;
  totalGastos: number = 0;
  chartData = {
    labels: [] as string[],
    datasets: [
      {
        label: 'Ingresos',
        data: [] as number[],
        fill: false,
        borderColor: '#42A5F5',
        tension: 0.4
      },
      {
        label: 'Gastos',
        data: [] as number[],
        fill: false,
```

App.component.css



```
You, hace 4 horas | 2 authors (You and one other) 4l3x1s456, hace 3 días • initial commit

import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { ChartComponent } from '../components/chart/chart.component';
import { CardComponent } from '../components/card/card.component';
import { InputNumberComponent } from '../components/input-number/input-number.component';
import { DataTableComponent } from '../components/data-table/data-table.component';
import { ToggleButtonComponent } from '../components/toggle-button/toggle-button.component';
import { CommonModule } from '@angular/common';

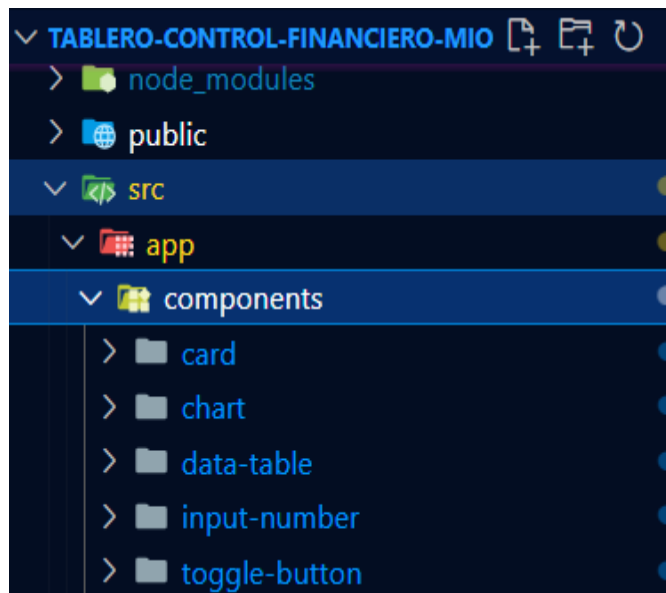
@Component({
  selector: 'app-root',
  imports: [RouterOutlet, ChartComponent, CardComponent, InputNumberComponent, DataTableComponent, ToggleButtonComponent],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  transactions: { date: string, type: string, amount: number }[] = [];
  totalIngresos: number = 0;
  totalGastos: number = 0;
  chartData = {
    labels: [] as string[],
    datasets: [
      {
        label: 'Ingresos',
        data: [] as number[],
        fill: false,
        borderColor: '#42A5F5',
        tension: 0.4
      },
      {
        label: 'Gastos',
        data: [] as number[],
        fill: false,
```

App.config.ts



```
app.config.ts M X
src > app > app.config.ts > ...
You, hace 4 horas | 2 authors (You and one other)
1 import { ApplicationConfig, provideZoneChangeDetection } from '@angular/core';
2 import { provideRouter } from '@angular/router';
3 import { providePrimeNG } from 'primeng/config';
4 import Aura from '@primeng/themes/aura';
5
6 import { routes } from './app.routes';
7 import { provideClientHydration, withEventReplay } from '@angular/platform-browser';
8
9 export const appConfig: ApplicationConfig = {
10   providers: [provideZoneChangeDetection({ eventCoalescing: true }),
11     provideRouter(routes), provideClientHydration(withEventReplay()),
12     providePrimeNG({
13       theme: {
14         preset: Aura
15       }
16     })
17   ],
18 };
19
```

Componentes



Historia de Usuarios



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

☐ Open 3 Closed 0

Author ▾ Labels ▾ Projects ▾ Milestones ▾ Assignees ▾ Types ▾ Newest ▾

☐ [Link components](#)
#3 · Ariel2006 opened 8 hours ago

☐ [Generate components](#) good first issue
#2 · 4l3x1s456 opened 8 hours ago

☐ [Install PrimeNG](#)
#1 · Ariel2006 opened 8 hours ago

Install PrimeNG #1

[Open](#) 0 / 1



Ariel2006 opened 8 hours ago

We open the terminal and send the commands to install the primeNG dependencies.



Sub-issues 0 of 1

[Comandos para instalar primeNG #4](#)

[Preview](#)

Create sub-issue ▾



Ariel2006 added a sub-issue now

[Comandos para instalar primeNG #4](#)

Pruebas Manuales

localhost:4200

TableroControlFinancieroMio

localhost:4200 dice
El valor no puede ser 0

Aceptar

Vista Resumen

Estado actual: Resumen

Fecha	Tipo	Monto

0

Agregar Ingreso

Agregar Gasto

Valor ingresado: 0

Procedencia: USD



7. Resultados

- Representa un grafico visual con los ingresos y gastos de diferentes formas.
- Se gestiona transacciones mediante tablas filtradas que se organizan.
- Su funcionalidad tiene vistas detalladas y resumidas que mejora la accesibilidad.
- Separación del código que asegura la mantenibilidad y accesibilidad.
- Link de github: <https://github.com/4l3x1s456/Web-Avanzada>

8. Conclusiones

- Con la estructura facilita la escalabilidad y que sea mantenible con el tablero, para que funcione estando en componentes individuales.
- PrimerNG permite integrar funcionalidades de forma rápida y eficiente para que mejore la experiencia del usuario.
- En el AppComponent se asegura que todos los componentes creados se sincronicen en tiempo real.

9. Recomendaciones

- Se deben agregar validaciones robustas en el InputNumberComponent para que se evite las entradas que son dobles o incorrectas.
- Se propone que se implementen animaciones para las transiciones con diferentes modos de vista que mejora la experiencia de usuario.
- Se incluye una personalización en los gráficos donde se ve que tipo específico de gastos.

10. Bibliografía o Referencias

- *PrimeNG - Angular UI component library.* (s/f). Primeng.org. Recuperado el 27 de enero de 2025, de <https://primeng.org/>
- *Angular.* (s/f). Angular.dev. Recuperado el 27 de enero de 2025, de <https://angular.dev/>