Trabajo Practico Diseño de Sistemas de la información

Alumno: Agustín Perez

**Introducción**

**Problema**

En la actualidad, muchas personas enfrentan dificultades para llevar un control eficiente de sus gastos. El uso de métodos tradicionales como libretas o planillas de cálculo puede generar desorganización, errores y falta de seguimiento. Esto impide por ejemplo poder saber bien en qué categorías se podría hacer una reducción de los gastos.

**Objetivos**

* Desarrollar una aplicación web que permita registrar, clasificar y visualizar gastos de forma simple e intuitiva.
* Facilitar al usuario el acceso a sus registros desde cualquier dispositivo con conexión a internet.
* Implementar mecanismos de validación y retroalimentación para asegurar la integridad de los datos ingresados.
* Aplicar buenas prácticas de modelado y desarrollo con herramientas modernas.

**Forma en la que se va a llevar a cabo el proyecto**

He decidido desarrollar este proyecto utilizando el Proceso Unificado de Desarrollo porque proporciona una estructura iterativa e incremental, que se adapta bien al tipo de proyecto que estamos realizando.

Esto me permite comenzar con una visión general del sistema y construir el producto en fases. Cada iteración facilita la validación con usuarios y la corrección de errores.

**Desarrollo**

**Primera iteración:**

Requisitos:

Funcionales:

* El sistema debe permitir al usuario iniciar sesión con credenciales válidas.
* El usuario debe poder agregar, editar, eliminar y visualizar gastos.
* Debe ofrecer un historial de gastos filtrado por fecha.
* Debe ofrecer un historial de gastos filtrado por categoría.
* Debe permitir cerrar sesión de forma segura.

No funcionales:

* La interfaz debe ser intuitiva y fácil de usar.
* El sistema debe responder en menos de 5 segundos ante cualquier acción.
* Los datos del usuario deben almacenarse de forma segura.
* Debe ser accesible desde distintos dispositivos.

Procesos identificados en el gestor de gastos:

* Iniciar sesión
* Registrar Usuario
* Agregar gastos
* Eliminar gastos
* Cerrar sesión
* Mostrar lista mensual de gastos
* Mostrar lista mensual por categoría de gastos
* Generar lista mensual de gastos
* Generar lista mensual por categoría de gastos
* Validar usuario
* Validar contraseña
* Guardar gasto en base de datos

Esquematización de los procesos “Iniciar Sesión, Registrar Usuario y Agregar Gasto”:

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama de casos de uso:

Diagrama

El contenido generado por IA puede ser incorrecto.Diagrama

El contenido generado por IA puede ser incorrecto.

Descripción de casos de uso Iniciar Sesión, Registrar usuario y Agregar Gasto:

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Diagramas de secuencia de los casos de uso Iniciar Sesión, Registrar usuario y Agregar Gasto:

Los siguientes diagramas de secuencia nos permiten visualizar como interactúan los componentes del sistema a lo largo del caso de uso.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Para el prototipado del sistema se utilizó una herramienta de IA. Durante el proceso de creación se tuvo en cuenta realizar un sistema sencillo y muy intuitivo para usuarios inexpertos donde la disposición de los botones, información y campos de relleno de información sea clara y no haya confusiones. La paleta de coleres se eligió como para que no sea molesta a la vista.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

La implementación del Sistema se realizó con React, una biblioteca de JavaScript, ya que ya la he utilizado antes en proyectos para otras materias como Desarrollo de Software. El desarrollo del Gestor todavía no está terminado, solo se codificaron los casos de uso mencionado y descriptos anteriormente.

Link a página web de la implementación: https://gestor-de-gastos-two.vercel.app/

Se realizaron 5 pruebas para descubrir posibles fallas en la implementación, en siguientes iteraciones se realizaran más:

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Segunda iteración:**

Arquitectura:

Se utiliza una arquitectura cliente – servidor donde:

* Cliente: Interfaz desarrollada en React que corre en el navegador.
* Servidor: API RESTful que procesa solicitudes y accede a la base de datos.
* Base de datos: Almacena usuarios, gastos y categorías.

Diagrama

El contenido generado por IA puede ser incorrecto.

Endpoints de la Api que permite comunicar al cliente con el servidor y este con la base de datos.

| **Acción** | **Método HTTP** | **Endpoint** | **Descripción** |
| --- | --- | --- | --- |
| Iniciar sesión | POST | /api/login | Autenticar usuario |
| Registrar usuario | POST | /api/register | Crear nuevo usuario |
| Agregar gasto | POST | /api/gastos | Registrar gasto nuevo |
| Editar gasto | PUT | /api/gastos/:id | Modificar un gasto existente |
| Eliminar gasto | DELETE | /api/gastos/:id | Eliminar un gasto |
| Ver lista de gastos | GET | /api/gastos?mes=05&categoria=transporte | Obtener gastos filtrados |

Clasificación de errores según su origen:

1. Errores funcionales: Ocurren cuando la funcionalidad no cumple con lo especificado.

* Iniciar sesión: El usuario no puede iniciar sesión con credenciales válidas

2. Errores de validación de datos: Se producen cuando se aceptan datos incorrectos o mal formateados.

3. Errores de interfaz: Problemas relacionados con la experiencia del usuario, no con la lógica del sistema.

4. Errores de comunicación cliente-servidor: Problemas en la conexión entre React y la API RESTful.

5. Errores de seguridad

* Se puede acceder a /api/gastos sin estar autenticado.

Clasificación de errores según su prioridad:

* Crítico
  + Impide el uso básico del sistema o pone en riesgo la seguridad o datos.
* Alto
  + Afecta funcionalidades importantes, pero tiene alguna solución temporal.
* Medio
  + Problema molesto, pero no impide el uso del sistema.
* Bajo
  + Detalles de estética, usabilidad o casos muy específicos.

Clasificación de errores encontrados en las pruebas hechas en la Primera Iteración:

Prueba 2: Agregar gasto (no muestra mensaje). Error tipo 3 y prioridad Medio.

Prueba 3: Agregar gasto con monto negativo. Error tipo 2 y prioridad Medio.

Prueba 4: Agregar Gasto con descripción infinita. Error tipo 2 y prioridad Alto.