

C#

Gestor de presupuesto

Objetivo

Desarrollar una aplicación para administración de presupuesto personal. La misma debe permitir crear y editar ingresos y egresos de dinero, y mostrar un balance resultante de las operaciones registradas.

Requerimientos Técnicos

Deberás utilizar Framework MVC. Los datos deben ser persistidos en una base de datos relacional. El esquema de la base puede armarse según se considere apropiado en base a los requerimientos del negocio. El trabajo realizado se subirá a un repositorio.

Pueden utilizarse plantillas para el frontend. La estructura de las vistas debe estar modularizada en layouts y secciones.

Secciones

Home

La pantalla de inicio deberá mostrar el balance actual, es decir, el resultante de los ingresos y egresos de dinero cargados, y un listado de los últimos 10 registrados.

ABM de operaciones (ingresos y egresos)

La aplicación deberá contener:

- Formulario de registro de operación. El mismo deberá contener:
 - Concepto
 - Monto
 - Fecha
 - Tipo (ingreso o egreso)

- Listado de operaciones registradas según su tipo (ingreso o egreso).
- Desde el listado, se debe poder modificar o eliminar una operación registrada previamente. No debe ser posible modificar el tipo de operación (ingreso o egreso) una vez creada.

Bonus

De forma adicional, podrías agregarle:

Autenticación de usuarios

Agregar un formulario de registro y login para permitir identificar al usuario que utiliza la aplicación, y vincular las operaciones registradas al usuario autenticado en el sistema, tanto para el listado y creación de nuevos registros. Los datos indispensables para permitir el ingreso deben ser un email y contraseña, pudiendo agregar los que se deseen.

Categorías de operaciones

Agregar la funcionalidad de categorizar las operaciones registradas en el gestor, como por ejemplo, una categoría “comida” para categorizar egresos. Adicionalmente, agregar la posibilidad de listar operaciones por categoría.

Utilización de JavaScript

Utilizar JavaScript para la comunicación entre las vistas y los Views

Utilización de Patrones de Diseño

Utilizar de posible el patrón de diseño Unit to Work para los repositorios.

Importante: no te preocupes si no podés completar todo, envialo de todas formas que te tendremos en cuenta igual.

Criterios a Evaluar

- El diseño debe ser responsive, pudiendo utilizarse CSS puro o algún framework de Frontend
- Código limpio, buenas prácticas de programación, en idioma inglés
- Correcto diseño de la base de datos
- Buenas prácticas de GIT: Commits declarativos y atomizados
- Buenas prácticas para el nombre de rutas