

CHALLENGE FULL STACK - JavaScript

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 desarrollar una API en **Node.js** junto a cualquiera de los siguientes **frameworks**, en sus versiones estables:

- [Express](#)
- [Adonis](#)
- [Koa](#)

En el caso de querer utilizar otro framework es posible, pero debe consultarse con anterioridad.

Los datos mostrados deben ser persistidos en una base de datos relacional. El esquema de datos puede armarse según se considere apropiado en base a los requerimientos del negocio. La API deberá exponer URLs que devuelvan datos en JSON.

Estos datos en JSON deberán ser consumidos por un cliente, a través de peticiones AJAX. El cliente puede ser armado con [React.js](#).

El trabajo realizado se subirá a un repositorio.

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, puede

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.

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