

EXAMEN TEMAS 12 Y 13

PARTE 1: MODO OFFLINE (6.5 PUNTOS)

EJERCICIO 1 (3 PUNTOS)

En esta ocasión vamos a realizar una App de Empleados. Como dicen que una imagen vale más que mil palabras, aquí tenéis la imagen de la pantalla de inicio:



Como puede observarse en la parte superior derecha, hay un botón agregar que nos cargará otra página mediante rutas para agregar al empleado



Cosas para hacer:

- El **modelo** (los datos los tenemos en la primera captura: nombre, apellidos y documento de tipo **string**, salario de tipo **number**)
- El **servicio**: guardará un **array** de tipo **modelo** los datos anteriores
- Los **componentes**: En este caso si es necesario crear, al menos 2 componentes. Uno para el listado y otro para el formulario,
- El **formulario agregar empleado** debe **controlar** que los campos sean todos **required**, y que tanto **documento** como **salario** sean **mayores** que **1**, en caso contrario no se agregarán al array de datos.
- Habrá una ruta desde el botón agregar al componente Agregar Empleado (se permite hacer desde el TS o desde el HTML).


EJERCICIO 2 (3.5 PUNTOS)

En este caso vamos a hacer una gestión de presupuesto:

En nuestra página de inicio debemos tener un **formulario** parecido a la siguiente (hay que comprobar que el presupuesto sea mayor o igual que 1 antes de pasar a la siguiente pantalla):



Formulario de ingreso de presupuesto. El título es "INGRESE PRESUPUESTO". Hay un campo de entrada de texto con un cursor visible y un botón "ACEPTAR" debajo.



Formulario de ingreso de presupuesto. El título es "INGRESE PRESUPUESTO". Hay un mensaje de error en rojo: "EL PRESUPUESTO DEBE SER MAYOR A CERO". El campo de entrada de texto contiene el valor "0" y hay un botón "ACEPTAR" debajo.

Ese valor debe ser accesible por la página siguiente, tenemos varias opciones para hacer esto: @Input o utilizando un servicio (recomendable esta opción)

En la siguiente pantalla nos aparecerá una ventana donde podemos ingresar un gasto y la cantidad de dicho gasto, a su vez tendremos el presupuesto total y el dinero que nos queda restante. Aquí tenemos una sugerencia de presentación.

The screenshot shows a web application titled "GASTOS". It is divided into two main sections: "Ingresar Gasto" (Add Expense) and "Listado de Gastos" (Expense List). In the "Ingresar Gasto" section, there is a text input field for "Gasto" with the placeholder text "Ej: Comida..", a numeric input field for "Cantidad" with the value "0", and a dark grey button labeled "AGREGAR GASTO". In the "Listado de Gastos" section, there are two summary boxes: a green box showing "Presupuesto: \$3000" and a grey box showing "Restante: \$3000".

Conforme vamos ingresando gastos se quedan reflejados en la parte inferior derecha y se descuentan del presupuesto.

This screenshot shows the same "GASTOS" application after a new expense has been added. In the "Ingresar Gasto" section, the inputs remain the same. In the "Listado de Gastos" section, the summary boxes are updated: "Presupuesto: \$3000" and "Restante: \$2000". Below these, a new entry is displayed: "Supermercado" with a dark grey pill-shaped button next to it containing the value "\$1000".

En caso de introducir un gasto de importe superior al restante mostrará un mensaje de error y no lo descontará del total

GASTOS

Ingresar Gasto

CANTIDAD INGRESADA ES MAYOR AL RESTANTE

Gasto

Nafta

Cantidad

3000

AGREGAR GASTO

Listado de Gastos

Presupuesto: \$3000

Restante: \$2000

Supermercado

\$1000

Para este ejercicio:

- Tendremos al menos 2 **componentes**, el de inicio y el de gastos. Cuando el importe sea correcto nos dirigiremos mediante rutas al segundo componente.
- Tendremos un **servicio** donde guardaremos el total del presupuesto y el dinero restante
- En este caso no necesitamos **modelo**

NOTA: En este punto hay que llamar a Juan para entregar estos 2 primeros ejercicios y que conecte Internet al equipo.