

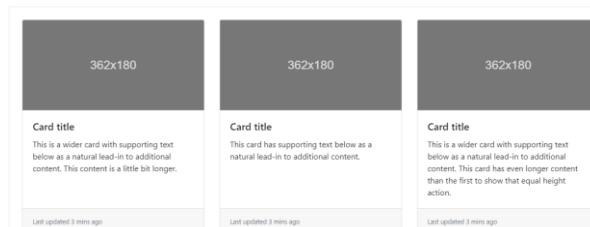
## PRÁCTICA BACKEND-FRONTEND

**OBJETIVO:** Poner en práctica los conceptos de: Arquitectura de Servicios Web con Express, establecimiento de Rutas, persistencia con MongoDB, testeo de API (Postman).

**PARTE2:** Adapte las plantillas y Componente que se referencia en cada punto para que pueda consumirse los datos desde el backend creado en la PARTE1 de la práctica

**PTO1)** Trabaje con los siguientes componentes:

- Utilice la plantilla para que el Slider muestre los **PRODUCTOS** destacados que provienen de la API en el backend, (consumir el servicio **GET-Recuperar los productos DESTACADOS**)
- Cree un nuevo componente que muestre un formulario para poder dar de alta los Productos en el formulario. (Consumir el servicio **“POST – Dar de alta un producto”**). VALIDE el ingreso de datos en el formulario.
- El servicio PUT y DELETE por el momento no serán consumidos en este punto.



Dar de alta un Producto (POST)	<b>Producto</b>
Recuperar TODOS los productos (GET)	nombre: String
Eliminar un producto (DELETE)	descripción: String
Modificar un producto (PUT)	imagen: String //url de una imagen para mostrar
Recuperar los productos DESTACADOS (GET)	precio: Number
	stock: Number
	destacado: Boolean // solo algunos productos son destacados

**PTO2)** Trabaje con los siguientes componentes:

- Basado en el la plantilla del formulario de conversión de divisas (PTO C – TP WEBSERVICE PUBLICOS) obtenido oportunamente, haga que al hacer click en el botón “CONVERTIR” luego de haber obtenido el resultado de la API publica de RapidAPI se consuma el servicio **POST: Dar de alta una Transacción.**
- Cree un nuevo componente que permita consumir los servicios webs: **GET: Recuperar Todas las Transacciones Registradas** del baken y mostrarlo en una lista y **GET: Recuperar TODAS las Transacciones que tengan como origen y destino las divisas (monedas) recibidas como parámetro.** Puede aplicar DOS botones Ej. “RECUPERAR TODOS” y otro “RECUPERAR POR FILTRO”.



### API de Referencia en el BAKCEND

-Dar de alta una Transaccion(POST) -Recuperar TODAS las Transacciones Registradas (GET) -Recuperar el histórico de transacciones de un determinado cliente (GET), utilizar email como clave -Recuperar TODAS las Transacciones que tengan como origen y destino las divisas (monedas) recibidas como parámetro (GET). Utilice el concepto de PARAMS. Nota: Ej. ARS-Argentina BRL-Brasil USD-EstadosUnid	<b>Transaccion</b> (representa el LOG de una operación) monedaOrigen: String cantidadOrigen: number monedaDestino: String cantidadDestino: number emailCliente: String tasaConversion: number // Dato que ingresa el cliente, y se ha utilizado para calcular cantidadDestino.
---	--



PTO3) Basado en el punto3 del Trabajo Práctico de Angular defina los siguientes componentes para la entidad TICKET. Para dar de alta un ticket tenga la precaución de haber dado de alta espectadores previamente (ej. consumiendo la API desde POSTMAN), **NO es necesario** implementar componentes para hacer el ABM de la entidad ESPECTADOR.

- un componente que muestre una lista de los tickets vendidos, para esto consuma:
  - o El servicio **GET: Recuperar todos los tickets**, indicando el espectador involucrado.
  - o El servicio **DELETE: Eliminar un ticket**.
  - o Para el servicio **GET: Recupere SOLO los tiket de determinada categoría de espectador** vea como aplicarlo sobre el componente según experiencia previa en pto2 donde también se consume web service con parámetros.
- un componente que muestre un formulario que VALIDE el ingreso de información y consuma:
  - o **POST: Dar de alta un ticket**. Se debería poder elegir el espectador que compre el ticket, ej desde listado.
  - o **PUT: Modificar un ticket**.

#### Referencia

Dar de alta una Espectador (POST) Obtener todas los Espectadores (GET) Obtener UN Espectador (GET)	<b>Espectador</b> apellido: String nombre: String dni: String email: String
- Dar de alta un ticket (POST), enviar el espectador como propiedad. - Recuperar TODOS los ticket (GET) incluyendo la información de los espectadores. - Eliminar un ticket (DELETE) - Modificar un ticket (PUT) -Recuperar SOLO los espectadores que tienen una determinada categoría de espectador //extranjero-local.	<b>Ticket</b> precioTicket: Number categoriaEspectador: String fechaCompra: String // gestinar fecha como string espectador: Espectador (Schema.Types.ObjectId) categoría del espectador puede ser: e = Extranjero, l = local.