



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).

PARTE1: Desarrolle una API que provea determinados servicios (rutas) sobre las siguientes situaciones problemáticas. En todos los casos debe testear todos los servicios de la API mediante alguna herramienta (ej. POSTMAN). Todos los puntos deben estar en **UN SOLO** proyecto (ej. tp5-apellidoynombre-backend).

PTO1) defina el siguiente esquema en el backend y cree una API que brinde los servicios abajo indicados:

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

PTO2) Defina el siguiente esquema de backend y cree una API que será de uso de una agencia de cambios que desea llevar un log de sus transacciones, la API brinda los servicios abajo indicados

-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	Transaccion (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 punto 3 del Trabajo Práctico de Angular defina el siguiente esquema en el backend y cree una API que brinde los servicios abajo indicados:

Dar de alta una Espectador (POST) Obtener todas los Espectadores (GET) Obtener UN Espectador (GET)	Espectador 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.	Ticket precioTicket: Number categoriaEspectador: String fechaCompra: String // gestinar fecha como string espectador: Espectador (Schema.Types.ObjectId) categoría del espectador puede ser: e = Extranjero, l = local.