

### Primer Bloque

- 1) Realizar maquetación del diseño suministrado y tomar en cuenta:
  - a) Todos los campos son requeridos.
  - b) El editar los campos de tarjetas, fecha y nombre modifica el diseño de la tarjeta.
  - c) El campo tarjeta solo puede contener números y un máximo de 16 caracteres.
  - d) El campo fecha de vencimiento debe tener formato mm/yy.
  - e) El campo fecha solo puede aceptar valores válidos para mes (01 a 12) y año (22 hasta año actual + 5).
  - f) El campo Nombre titular solo puede contener letras y letras con tildes y máximo de 20 caracteres.
  - g) En caso de que algún campo no sea válido se debe colocar texto en rojo debajo del campo indicando la causa de la invalidez.
- 2) Al pulsar botón de Agregar tarjeta.
  - a) Debe agregar la tarjeta a un bloque en el cual se mostrarán los campos de tarjeta, nombre y fecha vencimiento.
  - b) Para cada registro se debe contener un identificador único.
  - c) Debe validar que el formulario sea válido y que al no ser válido despliegue los textos en cada campo que no es válido.
  - d) El campo de número de tarjeta se debe mostrar enmascarado, solo mostrar los 12 primeros y 4 últimos dígitos (ej. 41\*\*\*\*\*1234).
- 3) Al pulsar botón Cancelar se deben limpiar todos los campos.

### Segunda Bloque

- 1) Crear RESTful API para el procesamiento de agregado/visualización de tarjetas.
  - a) Debe contener todos los métodos de CRUD. (aunque no los use todos)
  - b) Puede utilizar cualquier Web API de su preferencia (ej. Firebase, ASP.Net Core o MVC 5 en local host, Express, etc.)
  - c) Para la prueba no se tomará en cuenta ningún mecanismo de autenticación.
- 2) Conectar el primer bloque con el segundo bloque.
  - a) Al momento de agregar tarjeta esta debe ser almacenada en alguna base de datos o archivo de texto (ej. Firebase).
  - b) Lo único que se debe validar al momento de agregar es que los campos requeridos sean enviados.
  - c) Los API deben de devolver correctamente respuestas HTTP para cada caso, (ej. 404, 500, 302, 200, etc).

### **Criterios de Aceptación**

- 1) Colocar el proyecto en un repositorio ej. Github y compartir el enlace.
- 2) El proyecto debe tener una página de documentación en la cual se debe indicar.
  - a) Información de lo realizado.
  - b) Instrucciones de como correr y probar el proyecto.
  - c) Observaciones si se tienen.
- 3) Poseer al menos 70% realizado del primer bloque.
- 4) Poseer al menos un 50% realizado del segundo bloque.
- 5) Se debe indicar en la documentación los puntos realizados en cada bloque.

### **Notas**

- 1) Para el primer bloque puede utilizar cualquier framework de javascript si lo desea siempre y cuando se informe en la documentación.
- 2) Se evaluará la calidad y limpieza del código con más énfasis que la completitud total de los requerimientos.