

Bienvenido al test práctico!. A continuación presentamos el diseño y la descripción funcional de una pequeña aplicación que será la base del trabajo que deberás desarrollar. Una vez terminado, subila a github, y envíanos el link del repo con las instrucciones.

Tenés que usar el siguiente stack tecnológico para construir la aplicación:

- Cliente:
 - HTML/JS/CSS
 - Frameworks: NuxtJS, Vue o React (Deseable usar Nuxt)
 - Frameworks CSS o Componentes: Bootstrap

****** Para realizar el test se adjunta un mockup orientativo en cuanto a diseño. Se pueden utilizar distintas imágenes para backgrounds, logos, etc. Las imágenes de productos hay que utilizar las que llegan en las respuestas del api. Si algo no se entiende bien, mi correo es ncuevas@simplicar.com enviarme todas las consultas a esa casilla. Este test está enfocado en las tareas que harás en el día a día con el equipo.

Si vas a crear componentes, se espera que el nivel de atomización sea el necesario para resolver el problema (no mates una mosca con un cañón), simple es mejor que complejo.

Test Práctico

Contexto

La empresa tiene desarrollada una plataforma donde se puede gestionar Alta, Baja, Modificación de marcas y de todos sus productos. Entre los servicios que se brindan, uno es la generación de un sitio público donde los usuarios podrán descubrir dichos productos de la marca y comprarlos. La plataforma tiene implementado un API el cual va a proveer toda la información necesaria a las demás plataformas que quieran integrarse.

Se requiere

Armar una aplicación que consuma el API que provee la plataforma, para generar los sitios de las marcas (**debe ser responsive**).

La webapp debe contar al menos con dos vistas (**ver Mocks adjuntados**):

- HOME (**Obligatoria**): Punto de entrada al sitio, sólo informativo con un call to action que redirija al listado de productos.
- LISTADO DE PRODUCTOS (**Obligatoria**): En esta sección se deberá renderizar un listado de productos utilizando cards. Cada card deberá contar con imagen de producto, nombre o modelo y precio. Cuando el usuario haga click en la card debe redirigir a la página de detalle.
- DETALLE DE PRODUCTO (**No requerida, suma puntos**): Esta sección tendrá un carrusel con imágenes del producto,

Destacando por encima del mismo nombre o modelo, y precio (deberá estar en la columna izquierda). Además, debe contar con un formulario de consulta de disponibilidad del producto (deberá estar en la columna derecha). Los campos del formulario son email, número de teléfono, y Nombre (Todos son requeridos y agregar validación de email). La información debe enviarse al endpoint "lead".

API Endpoints

URL BASE: <https://4my1q6hsyo.api.quickmocker.com>

URL BASE de Imágenes: <https://s3.sa-east-1.amazonaws.com/simplimotos-stg.com>

HEADER REQUERIDO EN **TODOS LOS REQUESTS**:

"Authorization: Bearer qwertyuiopasdfghjklzxcvbnm123456"

GET /product

Respuesta:

Este endpoint retorna un listado de AUTOS.

```
[{
  "model": Modelo de auto,
  "year": Año del modelo,
  "amount": Precio del auto,
  "currency": Moneda,
  "gallery": array de imágenes,
  "detail": objeto con más información del vehículo,
  "specs": objeto características del vehículo,
}]
```

URL: <https://4my1q6hsyo.api.quickmocker.com/product>

Acciones Permitidas: GET

EJEMPLO:

```
curl -H 'Accept: application/json' -H "Authorization: Bearer
qwertyuiopasdfghjklzxcvbnm123456"
https://4my1q6hsyo.api.quickmocker.com/product
```

GET /product/{id}

Respuesta:

Este endpoint retorna la información de un auto según el id.

```
{
  "model": Modelo de auto,
  "year": Año del modelo,
  "amount": Precio del auto,
  "currency": Moneda,
  "gallery": array de imágenes,
  "detail": objeto con más información del vehículo,
  "specs": objeto características del vehículo,
}
```

URL: <https://4my1q6hsyo.api.quickmock.com/product/1>

Acciones Permitidas: GET

EJEMPLO:

```
curl -H 'Accept: application/json' -H "Authorization: Bearer
qwertyuiopasdfghjklzxcvbnm123456"
https://4my1q6hsyo.api.quickmock.com/product/1
```

GET /lead

Descripción:

Este endpoint se utiliza para inyectar la información del lead.

```
{
  "model": Modelo de auto,
  "year": Año del modelo,
  "amount": Precio del auto,
  "currency": Moneda,
  "gallery": array de imágenes,
  "detail": objeto con más información del vehículo,
  "specs": objeto características del vehículo,
}
```

URL: <https://4my1q6hsyo.api.quickmock.com/product/1>

Acciones Permitidas: POST

EJEMPLO:

```
curl -H 'Accept: application/json' -H "Content-Type: application/json" -H
"Authorization: Bearer qwertyuiopasdfghjklzxcvbnm123456"
-d '{"fullname": "Homer
Simpson", "email": "hjsimpson@gmail.com", "phone": "179287391", "product": "1"
```

```
}' -X POST https://4my1q6hsyo.api.quickmocker.com/lead
```