

Proyecto y Grupo



“Documento de relación entre Back-end y Front-end”

Hito: 2

Fecha entrega: 23-01-2017

Versión: 1

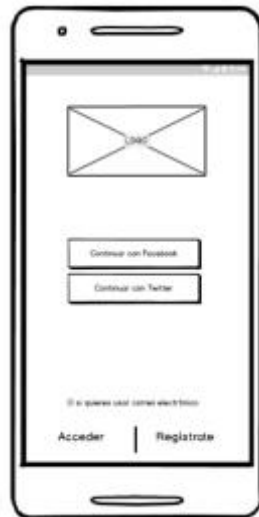
Componentes:

- **Emilio Maestre Hortal**
- **Jose Francisco Moreno Fernández**
- **Pablo López Riquelme**
- **Sergio Pérez Seré**

Documento de relación entre back-end y front-end

APLICACIÓN MÓVIL

Pantalla Principal



Al darle al botón de Facebook o Twitter se registrarán los usuarios en caso de que no estén en nuestra base de datos.

POST

```
/usuarios
```

Acceder



Al pulsar el botón de acceder enviamos el correo electrónico y la contraseña del usuario.

POST

```
/usuarios/login
```

Registro



Al darle al botón de Registro enviamos todos sus datos a nuestra API:

POST

```
/usuarios
```

Escaneo QR



El lector leerá el código QR de la tienda (que será su identificador y recogerá todos los datos de la tienda).

GET

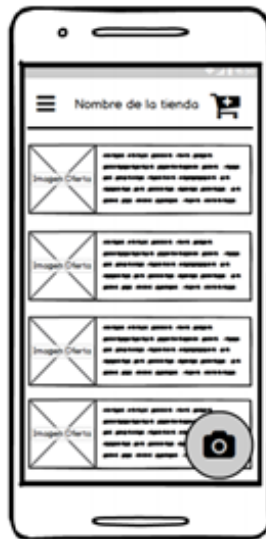
```
/tiendas?id={ID}
```

También asociaremos el usuario a una tienda, enviándole el id del usuario y el id de la tienda:

POST

```
/usuarios/usuario/tienda
```

Home



En esta pantalla se mostrarán las ofertas generales de productos en la tienda:

```
router.get('/ofertasProducto',comprobacionjwt,function(req,res){
});|
```

También se mostrarán las ofertas personalizadas de cada usuario:

```
router.get('/ofertasUsuario',comprobacionjwt,function(req,res){
});
```

Escaneo código de barras

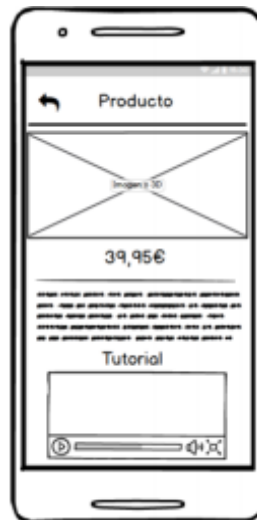


Recogemos los datos de un producto con su respectivo código de barras

GET

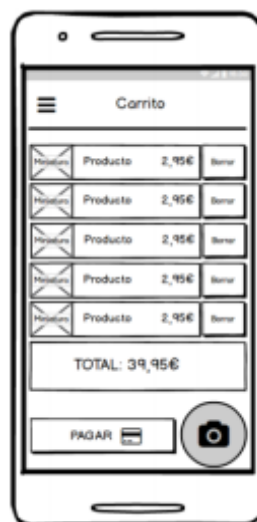
```
/productos?codigo={CODIGO}
```

Detalle de producto



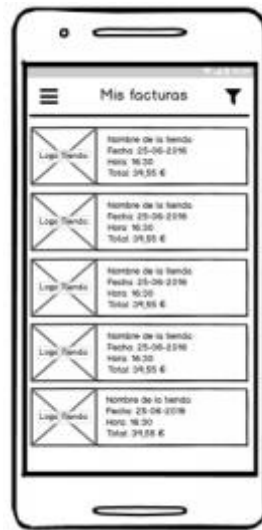
En esta página no se llama al back-end. Se recogen los datos obtenidos en la pantalla de escaneo de código de barras.

Carrito



En esta página no se llama al back-end. Estos datos se almacenan en local

Mis facturas



Aquí se reciben todas las facturas de un usuario haciendo la llamada de: facturas/

GET

```
/usuario?id={ID}
```

Detalle de factura



Se recogen todos los datos de una factura a través de su Id.

GET

```
/facturas?id={ID}
```

Ver tiendas



En esta página recibimos los datos de todas las tiendas que usen nuestro servicio:

GET

`/tiendas`

Cuando seleccionemos una tienda para ver sus datos no hará falta volver a llamar al back-end ya que realizaremos un filtro en el objeto que contiene los datos de todas las tiendas.

Perfil



En Mi Perfil recibimos todos los datos de un usuario. En este caso llamaremos a la ruta: **POST /loginjwt/info**. De esta forma no tenemos que volver a llamar a la base de datos ya que los datos los tenemos encriptados guardados en local. Solo enviamos por la cabecera el token encriptado y recibiremos los datos del usuario una vez el back-end lo desencripte

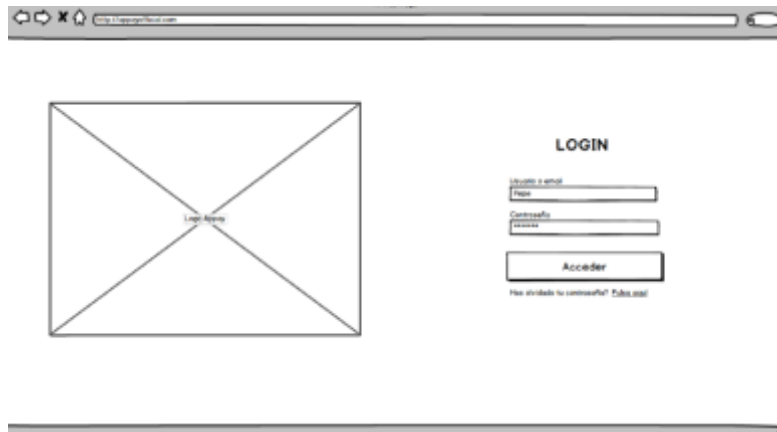
Al darle al botón de Guardar llamamos al Put de usuario enviándole los datos modificados del usuario:

PUT

/usuarios

PANEL DE ADMINISTRACIÓN

Login



Al pulsar el botón de acceder enviamos el correo electrónico y la contraseña del usuario.

POST

/usuarios/login

Home



En esta página llamaremos a toda la sección de estadísticas. **Estadísticas/**


```
//Esta funcion devuelve el numero de usuarios por tienda o totales
router.get('/usuariosTienda',comprobacionjwt,function(req,res){...
});
//Esta funcion devuelve el numero de registros los ultimos X dias, meses o años segun el parametro que se
router.get('/Registros',comprobacionjwt,function(req,res){...
});
//Esta funcion devuelve el numero de usuarios que han realizado una compra en un establecimiento
router.get('/usuariosFacturaTienda',comprobacionjwt,function(req,res){...
});
//Esta funcion devuelve el numero de facturas que se realizan con appay en un establecimiento
router.get('/usuariosTienda',comprobacionjwt,function(req,res){...
});
//Esta funcion devuelve el total de ingresos en facturas que se realizan con appay en un establecimiento
router.get('/usuariosTienda',comprobacionjwt,function(req,res){...
});
module.exports = router;
```

Perfil

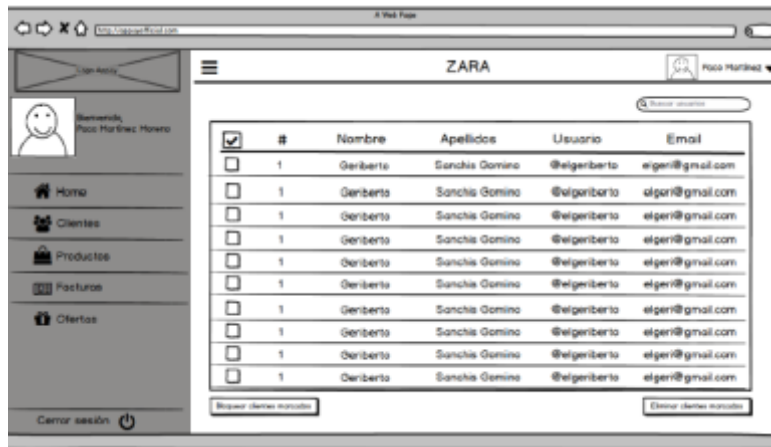
Al igual que en la App Movil, en 'Mi Perfil' recibimos todos los datos de un usuario. En este caso llamaremos a la ruta: **POST /loginjwt/info**. De esta forma no tenemos que volver a llamar a la base de datos ya que los datos los tenemos encriptados guardados en local. Solo enviamos por la cabecera el token encriptado y recibiremos los datos del usuario una vez el back-end lo desencripte

Al darle al botón de Guardar llamamos al Put de usuario enviándole los datos modificados del usuario:

PUT

/usuarios

Cientes



Desde esta página recogemos todos los clientes de una tienda:

GET

```
/tiendas?id={ID}
```

Detalles de Cliente



<input checked="" type="checkbox"/>	#	Día	Hora	Total
<input type="checkbox"/>	f1	13-12-2016	16:23	24,36€
<input type="checkbox"/>	f1	13-12-2016	16:23	24,36€
<input type="checkbox"/>	f1	13-12-2016	16:23	24,36€
<input type="checkbox"/>	f1	13-12-2016	16:23	24,36€
<input type="checkbox"/>	f1	13-12-2016	16:23	24,36€

Desplegable ofertas de usuario

Image	Nombre	Cod	Precio	Oferta	Inicio	Final
<input checked="" type="checkbox"/>	Salchicha	P7293H	4,95€	0%	??	??
<input checked="" type="checkbox"/>	Salchicha	P7293H	4,95€	0%	??	??
<input checked="" type="checkbox"/>	Salchicha	P7293H	4,95€	0%	??	??
<input checked="" type="checkbox"/>	Salchicha	P7293H	4,95€	0%	??	??

Leeremos los detalles del cliente mediante el token.

Para ver las facturas de un usuario usaremos: **facturas/**

GET

```
/usuario?id={ID}
```

Para ver las ofertas llamaremos al get de **/ofertasUsuarios**

Productos

<input checked="" type="checkbox"/>	Imagen	Nombre	Cod #	Precio	Oferta	Inicio	Final
<input type="checkbox"/>		Salchicha	P7243H	4,95€	0%	1.1.17	1.1.17
<input type="checkbox"/>		Salchicha	P7243H	4,95€	0%	1.1.17	1.1.17
<input type="checkbox"/>		Salchicha	P7243H	4,95€	0%	1.1.17	1.1.17
<input type="checkbox"/>		Salchicha	P7243H	4,95€	0%	1.1.17	1.1.17
<input type="checkbox"/>		Salchicha	P7243H	4,95€	0%	1.1.17	1.1.17
<input type="checkbox"/>		Salchicha	P7243H	4,95€	0%	1.1.17	1.1.17
<input type="checkbox"/>		Salchicha	P7243H	4,95€	0%	1.1.17	1.1.17
<input type="checkbox"/>		Salchicha	P7243H	4,95€	0%	1.1.17	1.1.17

Aquí listamos los productos de una tienda. Enviamos el id de la tienda como parámetro.

GET

```
/productos?id={ID}
```

Al final no utilizaremos el botón de actualizar producto, ya que las llamadas de modificación de producto se realizarán desde una ventana de detalle de producto

Detalle de Producto

En esta pantalla llamaremos al PUT del producto que recibirá los campos que se quieran modificar

PUT

```
/productos
```

También realizaremos una llamada al put de oferta de producto:

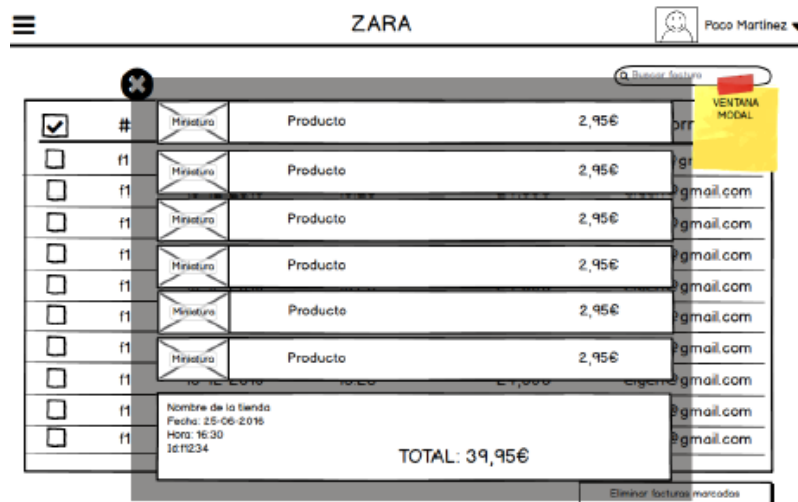
```
//PUT de OFERTAS DE PRODUCTO
router.put('/ofertasProducto',comprobacionjwt,function(req,res){ ...
});
```

Facturas

<input checked="" type="checkbox"/>	#	Dia	Hora	Total	Correo
<input type="checkbox"/>	f1	13-12-2015	16:23	24,35€	elgeri@gmail.com
<input type="checkbox"/>	f1	13-12-2015	16:23	24,35€	elgeri@gmail.com
<input type="checkbox"/>	f1	13-12-2015	16:23	24,35€	elgeri@gmail.com
<input type="checkbox"/>	f1	13-12-2015	16:23	24,35€	elgeri@gmail.com
<input type="checkbox"/>	f1	13-12-2015	16:23	24,35€	elgeri@gmail.com
<input type="checkbox"/>	f1	13-12-2015	16:23	24,35€	elgeri@gmail.com
<input type="checkbox"/>	f1	13-12-2015	16:23	24,35€	elgeri@gmail.com
<input type="checkbox"/>	f1	13-12-2015	16:23	24,35€	elgeri@gmail.com



```
/facturas
```

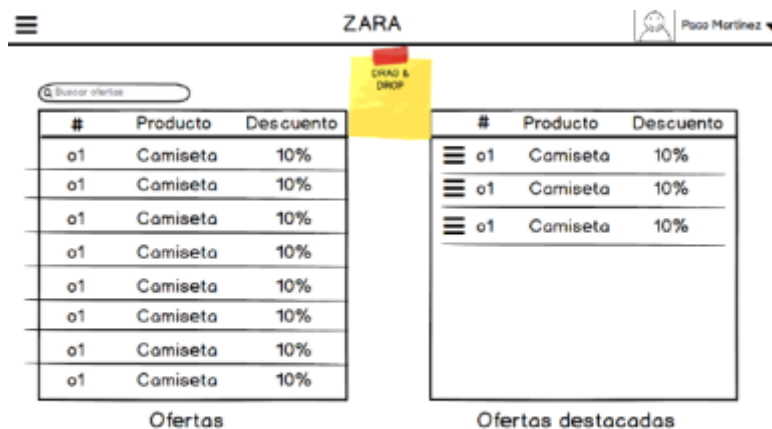


СЕТ



```
/facturas?id={ID}
```

of...



```
//PUT de OFERTAS DE PRODUCTO
router.put('/ofertasProducto',comprobacionjwt,function(req,res){...
});
```