

Santiago Gutiérrez Romero

I.E.S Punta del Verde 2ºASIR 21/2022

PROYECTO

Front-end

+

Back-end

MEAN:



Express

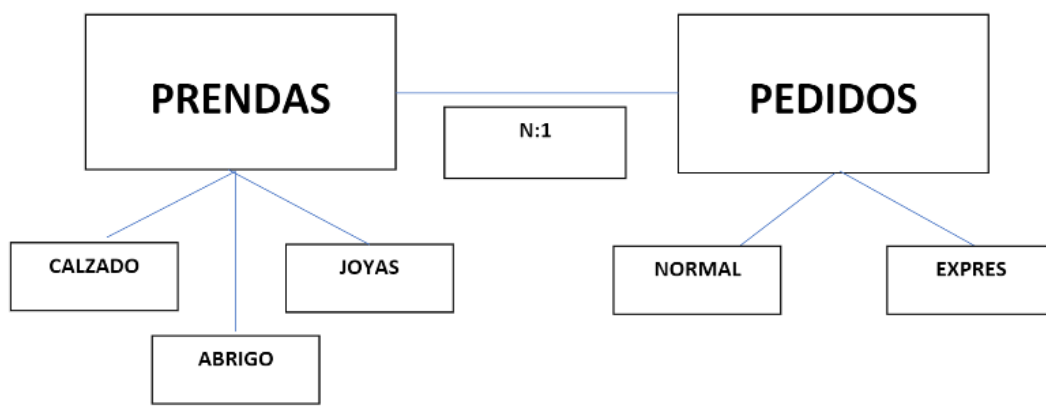


1.- INTRODUCCIÓN:

El siguiente proyecto ha sido desarrollado por Santiago Gutiérrez Romero, alumno de 2ºASIR del I.E.S. Punta del verde curso 21/2022:

El proyecto trata sobre una aplicación para la gestión de paquetería, particularmente se trataría de pedidos que llevan ropa.

Tanto los pedidos como la ropa en si tienen determinadas peculiaridades lo cual hace que hayamos creado diferentes clases con el fin de diferenciar a las diferentes entidades.



2.- API REST:

Para poner en funcionamiento la aplicación que hemos descrito anteriormente hemos usado una API REST en la cual hemos incluido las siguientes rutas:

```
//Obtener las prendas y su pedido correspondiente
this._router.get('/prendap', this.getPrends),
//Listar una prendas correspondiente a un pedido
this._router.get('/listarprend/:id', this.getPrendita)
//Listar una prenda
this._router.get('/prenda/:id', this.getPrenda),
//Obtener un usuario mediante ID
this._router.get('/unus/:usuario', this.getUnus)
//Añadir un pedido a la BD
this._router.post('/pedido', this.postPedido),
//Añadir una prenda a la BD
this._router.post('/prendas', this.postPrenda),
// Obtención del pedido con las prendas correspondientes
this._router.get('/pedidop', this.getPedido),
//Modificar un pedido
this._router.put('/ped/:id', this.updatePedido),
//Modificar un usuario
this._router.put('/actualizaUs/:usi/:_usuario/:_password', this.updateLog)
//Modificar el estado de un peddido
this._router.put('/estadoUs/:id/:estado', this.updateEst)
//Borrar una prenda
this._router.delete('/pedido/:_id', this.deletePrenda),
//Borrar un pedido y todas sus prendas
this._router.delete('/prenda/:_id', this.deletePedido),
//Enviar un correo
this._router.post('/enviar', this.sendMail),
//Obtener los usuarios de la BD
this._router.get('/validar', this.validate),
//Eliminar un usuario
this._router.delete('/usout/:_usuario', this.delus),
//Obtener los datos necesarios para el gráfico 3
this._router.get('/grafico3', this.getTotales),
//Obtener los datos para el gráfico 4
this._router.get('/grafico4', this.getDiferencias)
//Crear un usuario
this._router.post('/register', this.postUser)
```

3.- ESTRUCTURA DE DATOS EN BD

La estructura de los diferentes tipos de datos en la BD es la siguiente:

Colección prendas:

```
_id: 120
_tipoPrenda: "Abrigo"
_precioXmayor: 100
_precioPublico: 120
_fechaCompra: 2022-02-13T00:00:00.000+00:00
_material: "Cuero"
_paisFabric: "España"
_pedi: 3
_manga: "corto"
_cremallera: false
_cuello: "largo"
```

```
_id: 98
_tipoPrenda: "Joya"
_precioXmayor: 150
_precioPublico: 190
_fechaCompra: 9999-01-12T00:00:00.000+00:00
_material: "Plata"
_paisFabric: "Italia"
_pedi: 3
_quilates: 12
_peso: 5
```

```
_id: 555
_tipoPrenda: "Calzado"
_precioXmayor: 12
_precioPublico: 34
_fechaCompra: 2022-02-13T00:00:00.000+00:00
_material: "Cuero"
_paisFabric: "Italia"
_pedi: 3
_suela: "Goma"
_unidadesEnmercado: 45
_calidad: "Alto"
```

Colección pedidos:

```
_id: 1
_tipoPedido: "Normal"
_precioBase: 2
_diasAprox: 12
_fechaEnvio: 2001-04-30T22:00:00.000+00:00
_paisSalida: "España"
_estado: false
_incremento: 3
_impuesto: 3
_compania: "A"
```

```
_id: 2
_tipoPedido: "Expres"
_precioBase: 2
_diasAprox: 5
_fechaEnvio: 2001-03-31T22:00:00.000+00:00
_paisSalida: "Holanda"
_estado: false
_material: "Carton"
_volumen: 15
_proteccion: true
_compania: "B"
```

Colección usuarios:

```
_id: ObjectId("6207f689abd5b80b8eab6ac3")
_usuario: "root"
_password: "root"
```

4.- ESTRUCTURA DE LAS CLASES:

Prendas:

```
1  export class Prenda {
2      protected _id: number;
3      protected _precioXmayor: number;
4      protected _precioPublico: number;
5      private _fechaCompra: Date;
6      protected _material: string;
7      private _paisFabric: string;
8      private _pedi: number;
9
10     constructor(
11         id: number,
12         precioXmayor: number,
13         precioPublico: number,
14         fechaCompra: Date,
15         material: string,
16         paisFabric: string,
17         pedi: number) {
18
19         this._id = id;
20         this._precioXmayor = precioXmayor;
21         this._precioPublico = precioPublico;
22         this._fechaCompra = fechaCompra;
23         this._material = material;
24         this._paisFabric = paisFabric;
25         this._pedi = pedi;
26     }
27
28     // Metodos GET
```

Abrigos:

```
1  import { Prenda } from './prenda';
2
3  export class Abrigo extends Prenda {
4      private _manga: string;
5      private _cremallera: boolean;
6      private _cuello: string;
7      constructor(
8          id: number,
9          precioXmayor: number,
10         precioPublico: number,
11         fechaCompra: Date,
12         material: string,
13         paisFabric: string,
14         pedi: number,
15         manga: string,
16         cremallera: boolean,
17         cuello: string) {
18
19         super(id, precioXmayor, precioPublico, fechaCompra, material, paisFabric, pedi)
20         this._manga = manga;
21         this._cremallera = cremallera;
22         this._cuello = cuello;
23     }
24     //Metodos GET
25     get manga() {
26         return this._manga;
27     }
28     get cremallera() {
29         return this._cremallera;
30     }
31     get cuello() {
32         return this._cuello;
33     }
34     //METODOS CORRESPONDIENTES EN CADA UNA DE LAS SUBCLASES
```

Calzado:

```
import { Prenda } from './prenda';

export class Calzado extends Prenda {
  protected _suela: string;
  protected _unidadesEnmercado: number;
  protected _calidad: string;
  constructor(
    id: number,
    precioXmayor: number,
    precioPublico: number,
    fechaCompra: Date,
    material: string,
    paisFabric: string,
    pedi: number,
    suela: string,
    unidadesEnmercado: number,
    calidad: string) {

    super(id, precioXmayor, precioPublico, fechaCompra, material, paisFabric, pedi)
    this._suela = suela;
    this._unidadesEnmercado = unidadesEnmercado;
    this._calidad = calidad;
  }
}
```

Joya:

```
import { Prenda } from './prenda';

export class Joya extends Prenda {
  protected _quilates: number;
  protected _peso: number;
  constructor(
    id: number,
    precioXmayor: number,
    precioPublico: number,
    fechaCompra: Date,
    material: string,
    paisFabric: string,
    pedi: number,
    quilates: number,
    peso: number,
  ) {

    super(id, precioXmayor, precioPublico, fechaCompra, material, paisFabric, pedi)
    this._quilates = quilates;
    this._peso = peso
  }
}

//Metodos GET
```

Pedidos:

```
import { Prenda } from './prenda';

export class Pedido {
  protected _id: number;
  protected _precioBase: number;
  protected _diasAprox: number;
  private _compania: string;
  protected _fechaEnvio: Date;
  private _paisSalida: string;
  protected _estado: boolean;
  private _arrayPrendas: Array<Prenda>

  constructor(
    id: number,
    precioBase: number,
    diasAprox: number,
    compania: string,
    fechaEnvio: Date,
    paisSalida: string,
    estado: boolean) {

    this._id = id;
    this._precioBase = precioBase;
    this._diasAprox = diasAprox;
    this._compania = compania;
    this._fechaEnvio = fechaEnvio;
    this._paisSalida = paisSalida;
    this._estado = estado;
    this._arrayPrendas = new Array<Prenda>()
  }
}
```

Pedidos Normales:

```
1 import { Pedido } from './pedido';
2 import { Prenda } from './prenda';
3
4 export class Normal extends Pedido {
5   protected _incremento: number;
6   protected _impuesto: number;
7
8   constructor(
9     id: number,
10    precioBase: number,
11    diasAprox: number,
12    compania: string,
13    fechaEnvio: Date,
14    paisSalida: string,
15    estado: boolean,
16    incremento: number,
17    impuesto: number) {
18
19     super(id, precioBase, diasAprox, compania, fechaEnvio, paisSalida, estado)
20
21     this._incremento = incremento;
22     this._impuesto = impuesto;
23   }
}
```


Pedidos Express:

```
import { Pedido } from './pedido';
import { Prenda } from './prenda';

export class Expres extends Pedido {
  protected _material: string;
  protected _volumen: number;
  protected _proteccion: boolean;

  constructor(
    id: number,
    precioBase: number,
    diasAprox: number,
    compania: string,
    fechaEnvio: Date,
    paisSalida: string,
    estado: boolean,
    material: string,
    volumen: number,
    proteccion: boolean,
  ) {
    super(id, precioBase, diasAprox, compania, fechaEnvio, paisSalida, estado);

    this._material = material;
    this._volumen = volumen;
    this._proteccion = proteccion;
  }
}
```

Recibir enviar datos:

```
export class Todos {
  private _totalIAN: any;
  private _totalJN: number;
  private _totalCA: number;
  private _totalAE: number;
  private _totalJE: number;
  private _totalCE: number;

  constructor(
    _totalIAN: any,
    _totalJN: number,
    _totalCA: number,
    _totalAE: number,
    _totalJE: number,
    _totalCE: number,
  ) {
    this._totalIAN = _totalIAN;
    this._totalJN = _totalJN;
    this._totalCA = _totalCA;
    this._totalAE = _totalAE;
    this._totalJE = _totalJE;
    this._totalCE = _totalCE;
  }
}
```

```
export class User {
  protected _id: any;
  protected _usuario: string;
  protected _password: string;

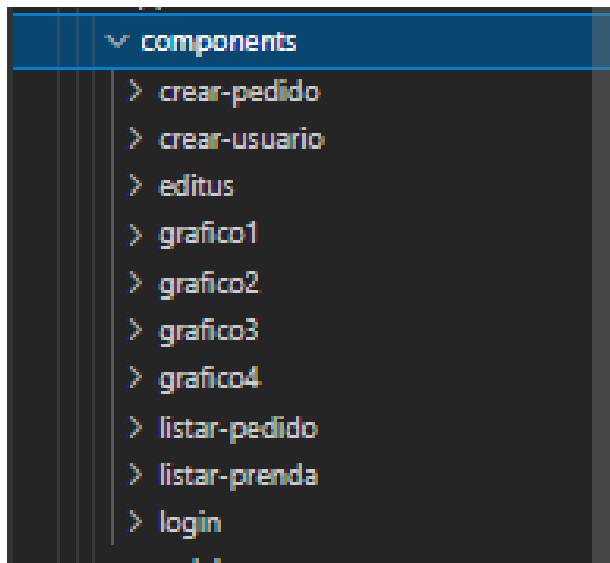
  constructor(
    id: any,
    usuario: string,
    password: string, ) {
    this._id = id;
    this._usuario = usuario;
    this._password = password;
  }
}
```

```
export class PrecioCl {
  private _idPed: number;
  private _precio: number;

  constructor(
    _idPed: number,
    _precio: number,
  ) {
    this._idPed = _idPed;
    this._precio = _precio;
  }
}
```

5.-COMPONENTES DEL FRONT END:

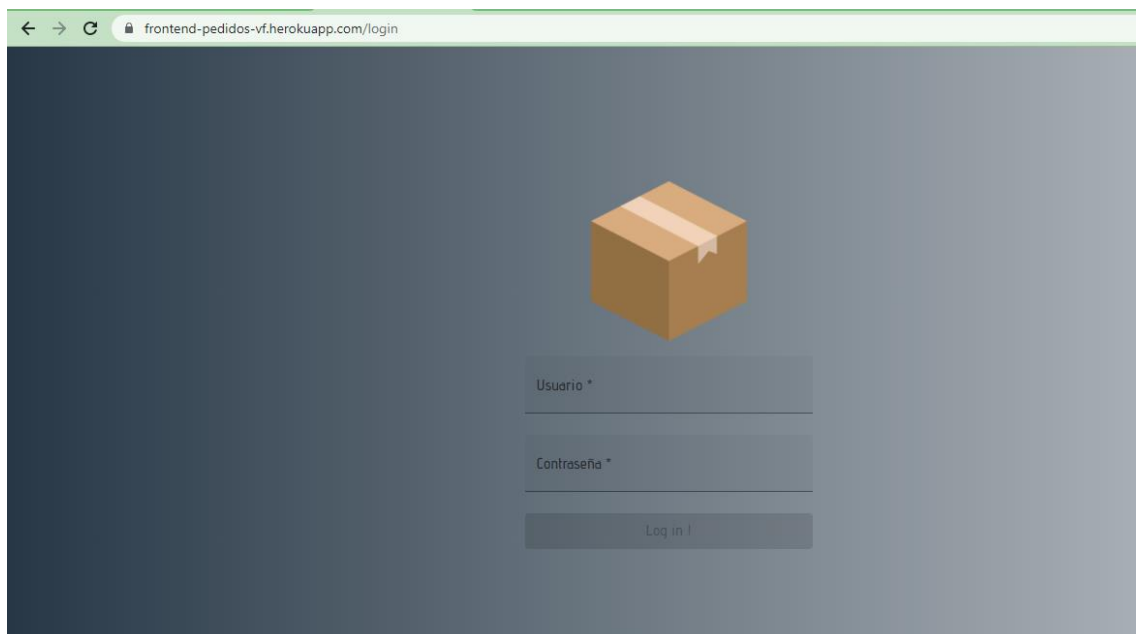
A continuación, se pueden ver los diferentes componentes que están implicados en el desarrollo angular de la aplicación:



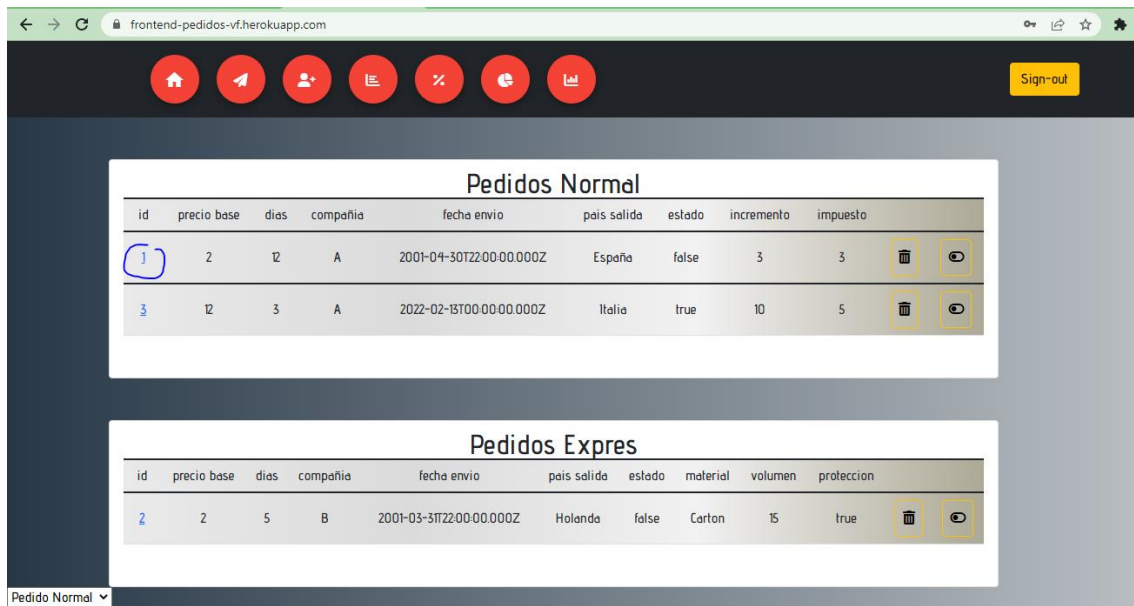
6.- RUTA GUIADA POR LA APLICACIÓN:

Ruta de inicio:

<https://frontend-pedidos-vf.herokuapp.com/login>



Accedemos con user:root pass:root



Accedemos al pedido 1 y creamos un abrigo:



Podemos eliminarlo:

frontend-pedidos-vf.herokuapp.com/listarprend/1

Sign-out

ABRIGOS

id	Por Mayor	Venta al Publico	fecha Compra	material	pais fabric	manga	cremallera	cueillo
12300	34	78	2022-02-18T00:00:00.000Z	Cuero	España	corta	true	corta

Añadir Abrigo al pedido

id de la prenda

Precio al por mayor

Precio venta al publico

dd/mm/aaaa

Material de fabricación

País de fabricación

Tipo de Manga del abrigo

Cremallera

Tipo de cuello del abrigo

Añadir

Ahora nos movemos a la pestaña de envío de correos:

frontend-pedidos-vf.herokuapp.com/listarprend/1

Sign-out

Añadir Abrigo al pedido

id de la prenda

Precio al por mayor

Precio venta al publico

dd/mm/aaaa

Material de fabricación

País de fabricación

Tipo de Manga del abrigo

Cremallera

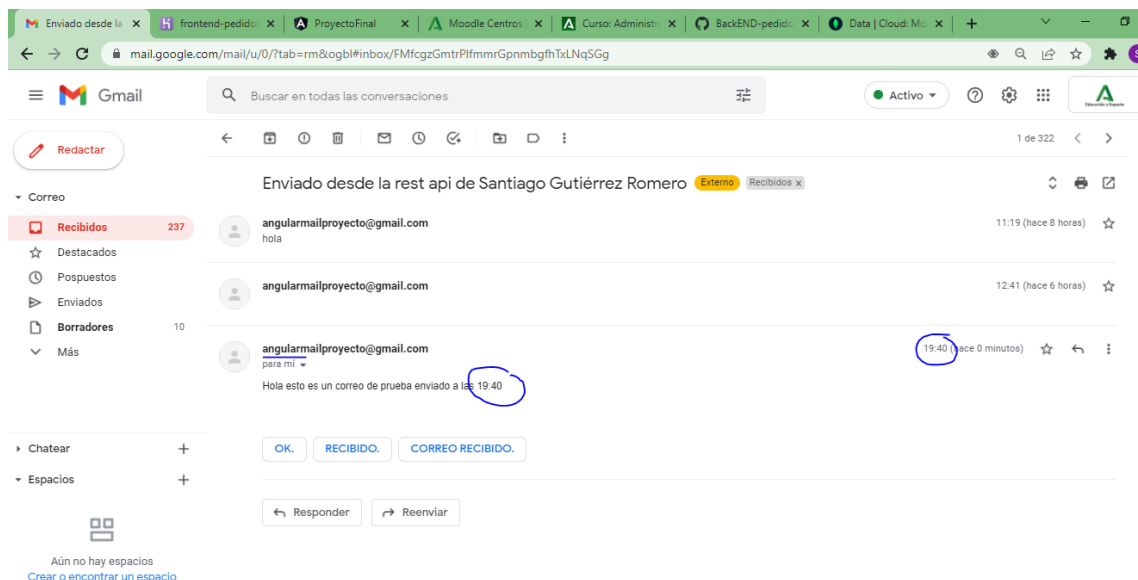
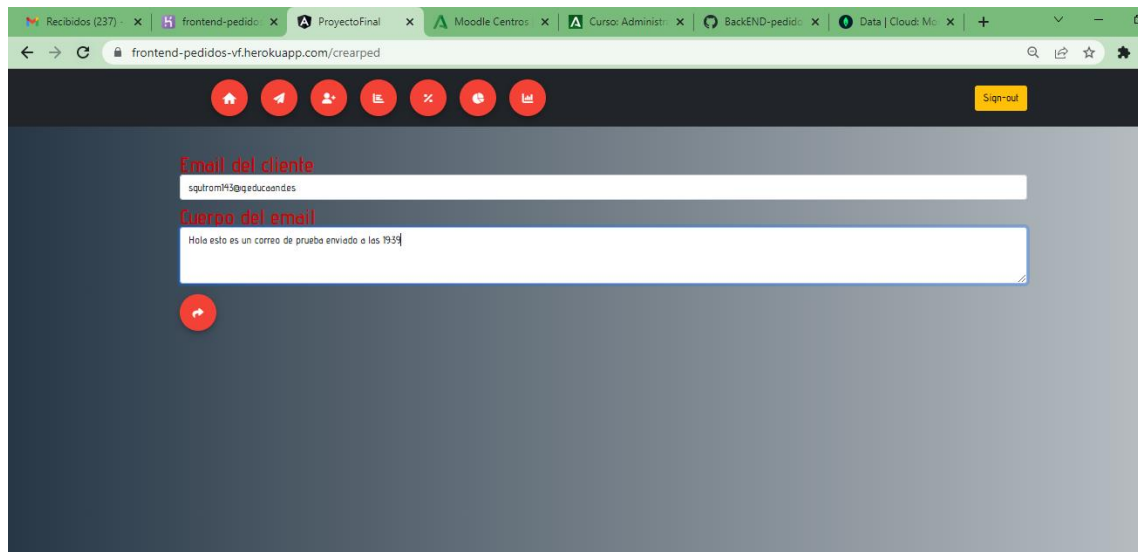
Tipo de cuello del abrigo

Añadir

JOYAS

id	Por Mayor	Venta al Publico	fecha Compra	material	pais fabric	quiletes	peso
12339	12	35	2022-02-12T00:00:00.000Z	Visulteria	España	5	10

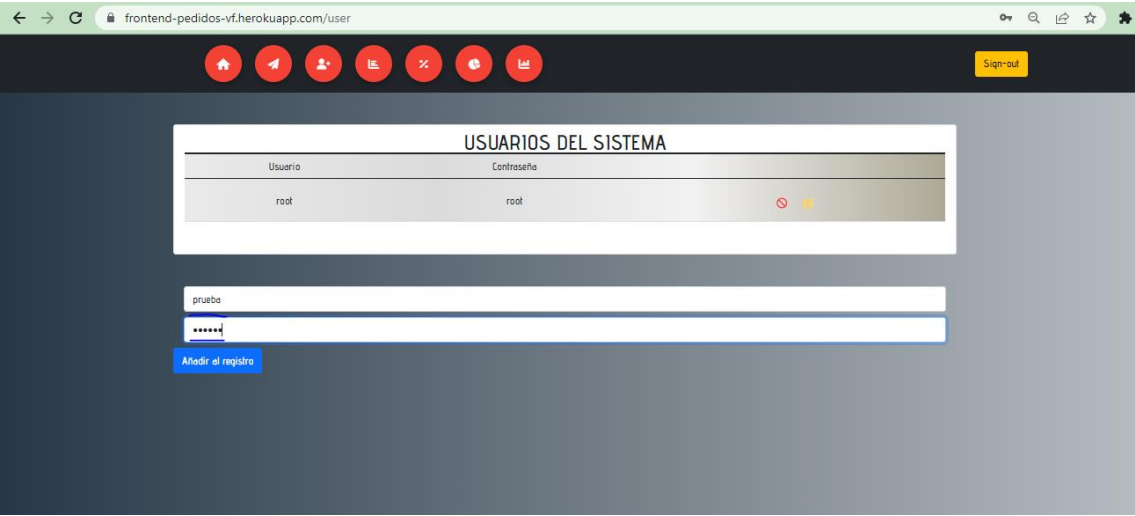
Enviamos un correo y comprobamos que llega:



Nos movemos a la pestaña de creación de usuarios:



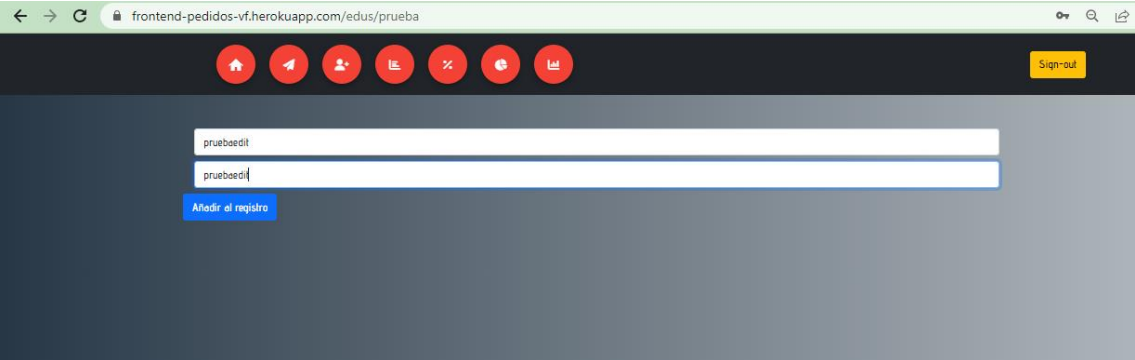
Creamos el usuario prueba:



Ahora podremos acceder con el a la aplicación. Vamos a editarlo:

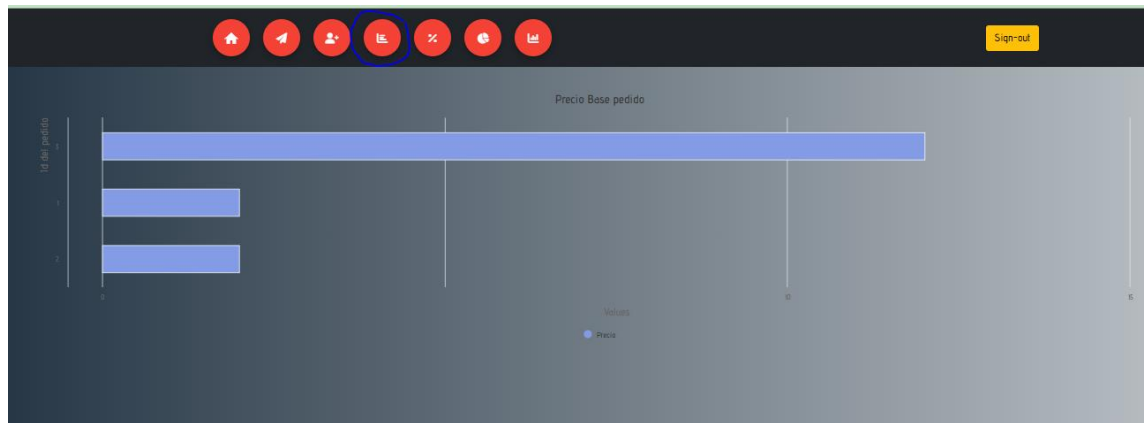


Cambiamos usuario y contraseña:

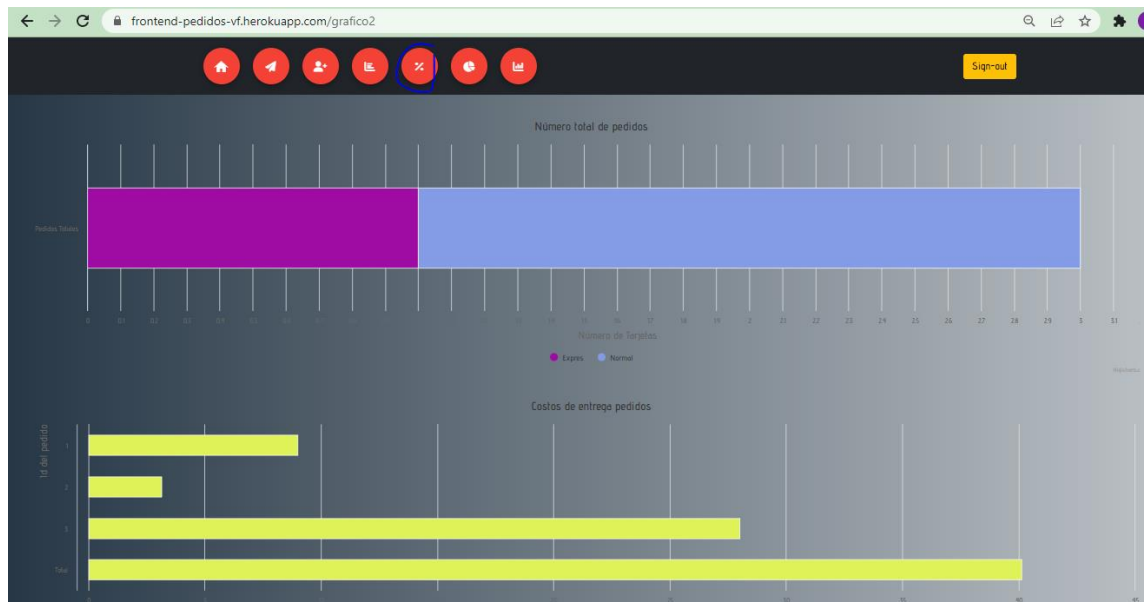




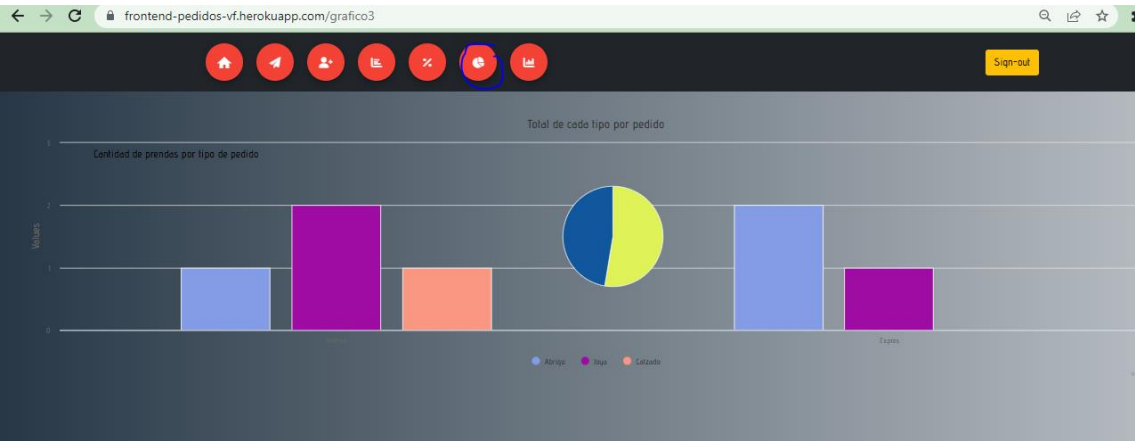
Nos movemos a la gráfica 1:



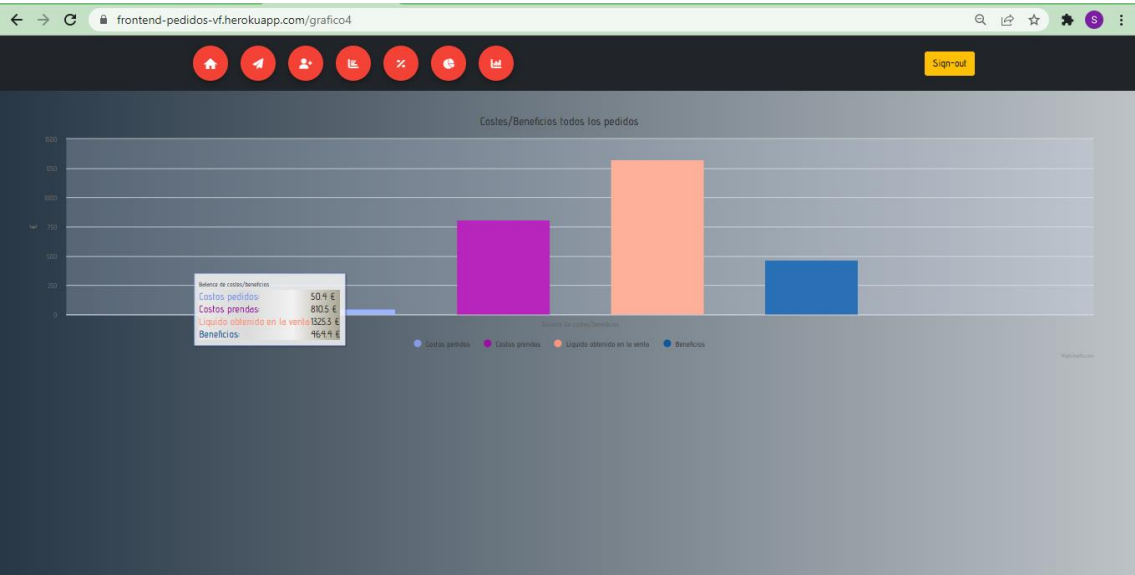
Nos movemos a la gráfica 2



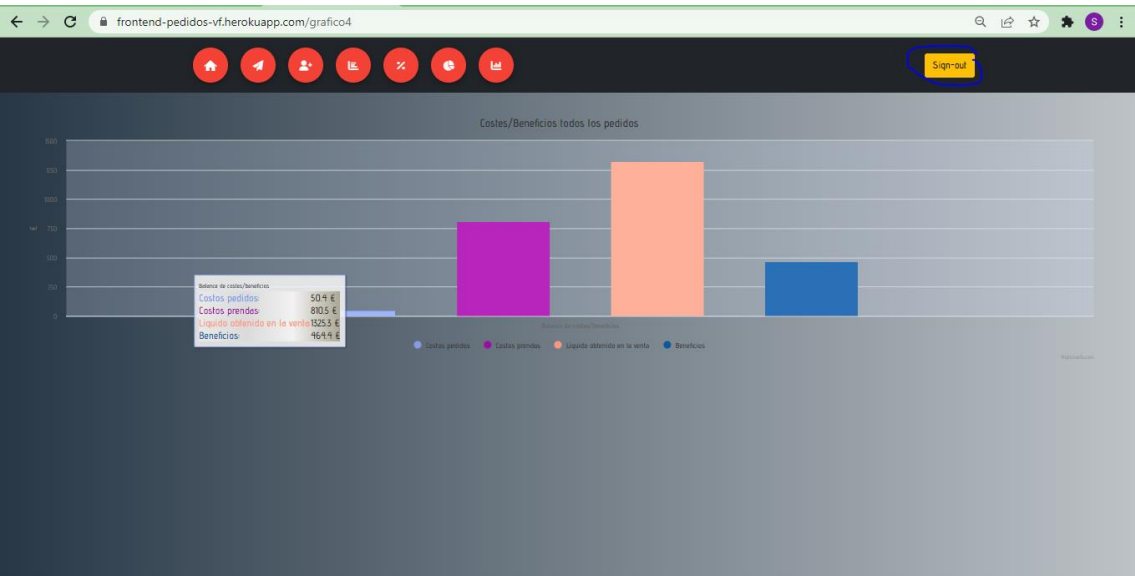
Nos movemos a la gráfica 3:

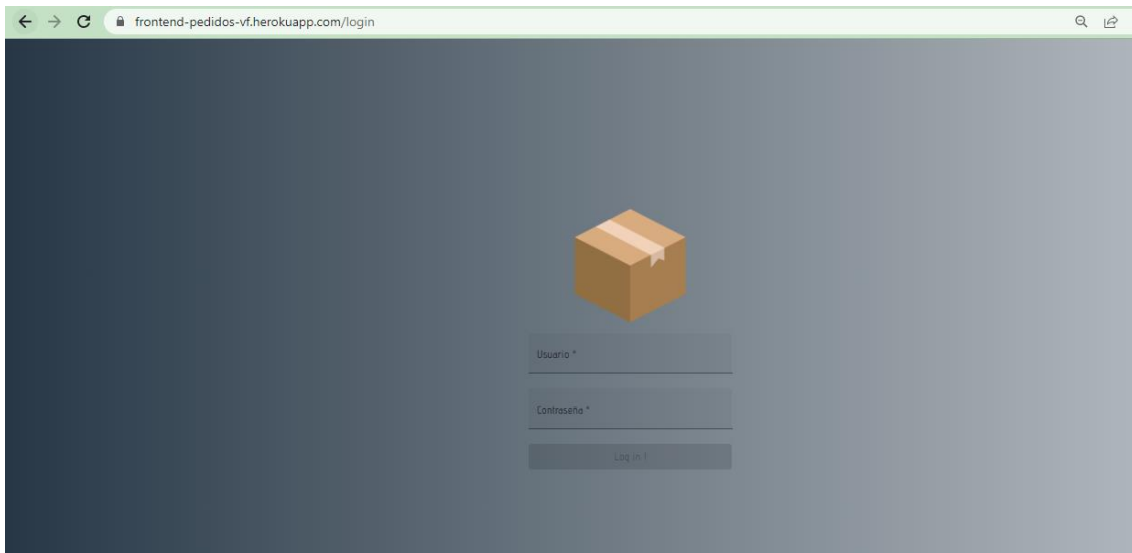


Nos movemos a la gráfica 4:



Salimos de la aplicación:





6.- SERVICIOS DE COMUNICACIÓN:

```
proyectoFinal > src > app > services > TS login.service.ts > LoginService > url
1  import { HttpClient } from '@angular/common/http';
2  import { Injectable } from '@angular/core';
3
4  @Injectable({
5    providedIn: 'root'
6  })
7  export class LoginService {
8    private url = "https://backend-pedidos-vf.herokuapp.com"
9    constructor(private http: HttpClient) {}
10
11    getUsers() {
12      const url = `${this.url}/validar`;
13      return this.http.get(url);
14    }
15
16    getUnUs(usuario: any) {
17      const url = `${this.url}/unos/${usuario}`;
18      return this.http.get(url);
19    }
20
21    editatUs(usi:string,_usuario:string,_password:string){
22      return this.http.put(this.url + '/actualizaUs/' + usi + '/' + _usuario + '/' + _password, {responseType: 'text'});
23    }
24
25    eliminarUsuario(_usuario: string) {
26      //const url = `${this.url}/usout/${_usuario}`;
27      return this.http.delete(this.url + '/usout/' + _usuario, {responseType: 'text'});
28    }
29
30    addUsuario(doc: any) {
31      const url = `${this.url}/register`;
32      return this.http.post(url, doc);
33    }
34  }
35
36
```

```

1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Observable } from 'rxjs';
4
5  @Injectable({
6    providedIn: 'root'
7  })
8  export class PedidoService {
9    private url = "https://backend-pedidos-vf.herokuapp.com"
10    constructor(private http: HttpClient) {}
11
12    getPedBasic() {
13      const url = `${this.url}/pedidop`;
14      return this.http.get(url);
15    }
16
17    addPedido(doc: any) {
18      const url = `${this.url}/pedido`;
19      return this.http.post(url, doc);
20    }
21
22    eliminarPedido(_id: number) {
23      //const url = `${this.url}/pedido/${_id}`;
24      return this.http.delete(this.url + '/pedido/' + _id, {responseType: 'text'});
25      //return this.http.delete(this.url + '/usout/' + _usuario, {responseType: 'text'});
26    }
27
28    editaEst(id:any,estado:any){
29      return this.http.put(this.url + '/estadoUs/' + id + '/' + estado, {responseType: 'text'});
30    }
31    getDametodo() {
32      return this.http.get(this.url + '/grafico3', {responseType: 'text'});
33    }
34
35    getDameNum() {
36      return this.http.get(this.url + '/grafico4', {responseType: 'text'});
37    }
38
39    enviarCorreo(doc: any) {
40      const url = `${this.url}/enviar`;
41      console.log("Esto es el servicio");
42      console.log(doc);
43      return this.http.post(url, doc);
44    }
45  }
46
47

```

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class PrendaService {
8
9   private url = "https://backend-pedidos-vf.herokuapp.com"
10   constructor(private http: HttpClient) { }
11
12   getPrenda(id: any) {
13     const url = `${this.url}/listarprend/${id}`;
14     return this.http.get(url);
15   }
16
17   addPrendas(doc: any) {
18     const url = `${this.url}/prendas`;
19     return this.http.post(url, doc);
20   }
21
22   eliminarPrenda(_id: number) {
23     //const url = `${this.url}/prenda/${_id}`;
24     //return this.http.delete(this.url + '/pedido/' + _id , {responseType: 'text'});
25     return this.http.delete(this.url + '/prenda/' + _id, {responseType: 'text'});
26   }
27
28 }
29
```