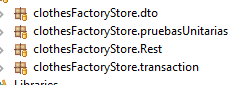
**ClothesFactoryStore**

**Backend - Documentación técnica**

Dentro de la aplicación **ClothesFactoryStoreEAR.ear,** se encuentra empaquetadala aplicación **ClothesFactoryStore.war**

La aplicación cuenta con 4 paquetes:



**clothesFactoryStore.dto** = Objetos de transferencia de datos utilizados en toda la aplicación.

**clothesFactoryStore.pruebasUnitarias** = Pruebas unitarias.

**clothesFactoryStore.transaction** = Core de la aplicación.

**clothesFactoryStore.Rest =** Expone los servicios REST, los cuales son los siguientes:

* [http://IPHOST:PUERTO/ClothesFactoryStore/RestWs/Pedidos/**getTiposPrenda**](http://IPHOST:PUERTO/ClothesFactoryStore/RestWs/Pedidos/getTiposPrenda)

**Método GET**

* + Obtiene los tipos de prenda, no recibe parámetros.
    - Ejemplo Respuesta:

{

"listaItemListaDTO": [

{

"codigo": "1",

"nombre": "Camisas Formales",

"labelMostrar": "Camisas Formales",

"estado": true

},

{

"codigo": "2",

"nombre": "Pantalones formales",

"labelMostrar": "Pantalones formales",

"estado": true

},

{

"codigo": "3",

"nombre": "Chaquetas",

"labelMostrar": "Chaquetas",

"estado": true

}

],

"nombreLista": "tiposPrenda"

}

* [http://IPHOST:PUERTO/ClothesFactoryStore/RestWs/Pedidos/**getCiudadesEnvio**](http://IPHOST:PUERTO/ClothesFactoryStore/RestWs/Pedidos/getTiposPrenda)

**Método GET**

* + Obtiene las ciudades disponibles para el envío del pedido y el valor del envío a la ciudad, no recibe parámetros.
    - Ejemplo Respuesta:

[

{

"codigo": 1,

"nombre": "Bogotá",

"costoEnvio": 35000

},

{

"codigo": 2,

"nombre": "Mosquera",

"costoEnvio": 45000

},

{

"codigo": 3,

"nombre": "Chia",

"costoEnvio": 50000

},

{

"codigo": 4,

"nombre": "Tunja",

"costoEnvio": 12000

}

]

* [http://IPHOST:PUERTO/ClothesFactoryStore/RestWs/Pedidos/**getPrendasVenta**](http://IPHOST:PUERTO/ClothesFactoryStore/RestWs/Pedidos/getTiposPrenda)

**Método GET**

* + Obtiene las prendas disponibles para la venta, no recibe parámetros.
    - Ejemplo Respuesta:

[

{

"id": 1,

"costoSinEnvio": 40000,

"costoConEnvio": 0,

"idTipo": 1,

"existenciasEnInventario": true,

"nombre": "CAMISA MANGA LARGA",

"tallasDisponibles": [

"S",

"M",

"L"

],

"coloresDisponiles": [

"AZUL",

"ROJO",

"VERDE"

]

},

{

"id": 2,

"costoSinEnvio": 55000,

"costoConEnvio": 0,

"idTipo": 2,

"existenciasEnInventario": true,

"nombre": "PANTALON CORTO",

"tallasDisponibles": [

"S",

"M",

"L"

],

"coloresDisponiles": [

"AZUL",

"ROJO",

"VERDE"

]

},

{

"id": 3,

"costoSinEnvio": 100000,

"costoConEnvio": 0,

"idTipo": 3,

"existenciasEnInventario": true,

"nombre": "CHAQUETA",

"tallasDisponibles": [

"S",

"M",

"L"

],

"coloresDisponiles": [

"AZUL",

"ROJO",

"VERDE"

]

}

]

* [http://IPHOST:PUERTO/ClothesFactoryStore/RestWs/Pedidos/**setPedido**](http://IPHOST:PUERTO/ClothesFactoryStore/RestWs/Pedidos/getTiposPrenda)
* **Método POST**
  + Recibe la información del pedido, recibo un objeto como parámetro, el código de respuesta exitoso es 0, 1 Error.
    - Ejemplo Solicitud:

{

"nombreCliente":"Felipe",

"documento":"1069733998",

"tipoDocumento":"CC",

"costoTotalConEnvio":50000,

"costoTotalSinEnvio":50000,

"direccion":"Cra 2b numero 25 a 16",

"ciudadEnvioDTO":{

"codigo":1,

"costoEnvio":35000,

"nombre":"Bogotá"

},

"listaItemPedidoDTO":[

{

"idPrendaDTO":1,

"nombrePrendaDTO":"CAMISA",

"cantidad":1,

"color":"AZUL",

"talla":"M",

"costoSinEnvio":54000

}

]

}

* + - Ejemplo Respuesta:

{

"codigoTransaccion": 0,

"mensaje": "Pedido Registrado Exitosamente"

}

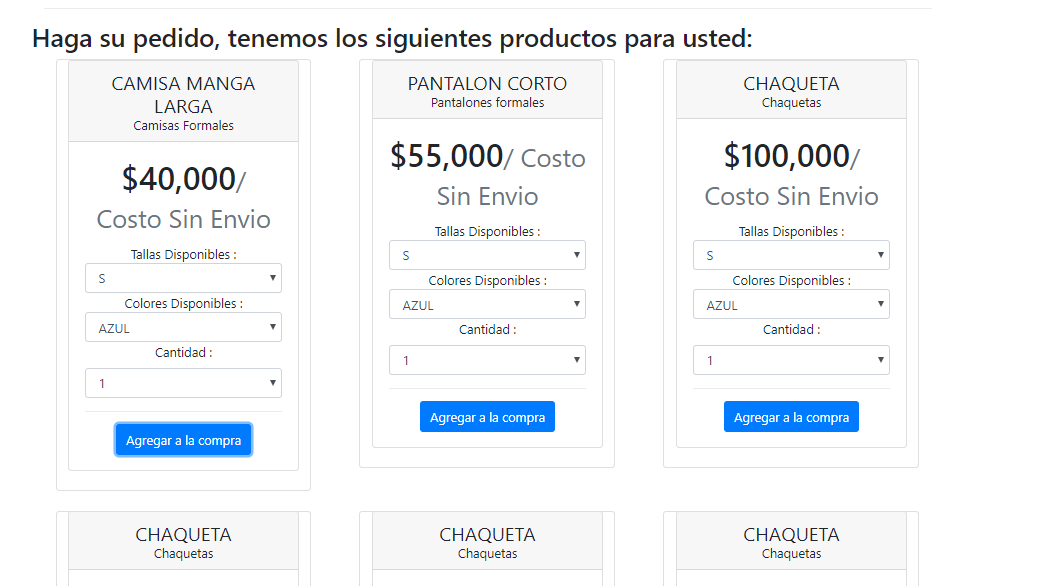
**Front – End**

La aplicación esta configurada para que si falla el consumo del servicio Rest, se muestren productos y opciones quemadas en el Front.

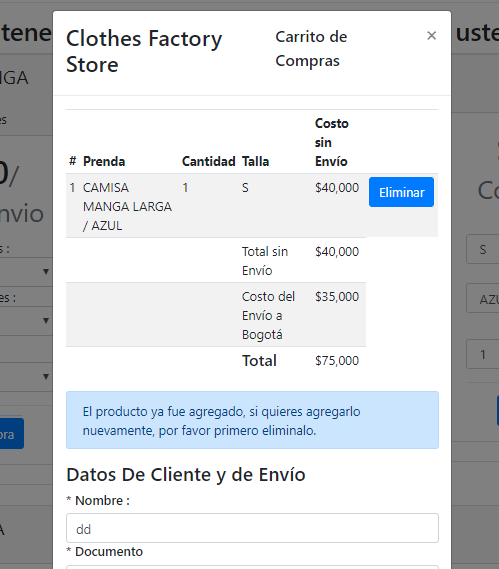
Dentro del controlador (**js/controllers/clothesFactoryStoreController.js**), podemos modificar la IP y el puerto del servidor donde se consumen los servicios REST:

***$scope.url = "http://localhost:9089";***

El usuario selecciona la talla, el color y la cantidad requerida y da click en el Botón “Agregar a la compra”

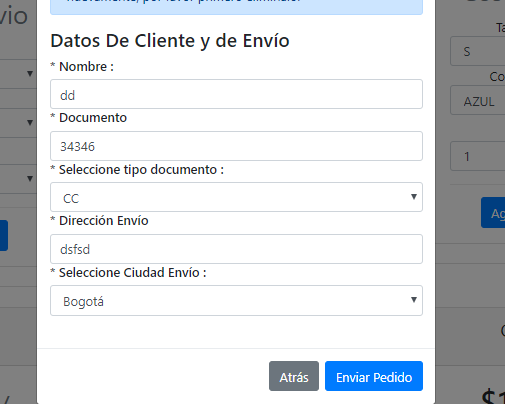


De inmediato se despliega el carrito de compras, como se ve en la siguiente imagen:



El usuario puede borrar, y agregar máximo 5 prendas

El usuario debe diligenciar el formulario que se ve a continuación:





Al dar Click en Enviar Pedido la información viajara al backend.

**Pruebas Unitarias**

Adjunto evidencias de las pruebas unitarias realizadas:

