

app.component.html

```
<router-outlet></router-outlet>
```

```

<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-body">
                <p-table #dtable [value]="cabeceras" [columns]="selectedColumns" [tableStyle]="
[paginator]="true" [rows]="10"
[globalFilterFields]="['tipo', 'numero_fac',
[(selection)]=\"selectedCabeceras\" (selectionChange)=\"onSelectionChange($event)\">

pTemplate=\"caption\">
                <div class=\"p-d-flex justify-content-between align-items-center\">
                    [(ngModel)]=\"selectedSucursal\" (onChange)=\"onSucursalChange($event)\"
                    dropdown>
                        <p-multiSelect [options]=\"cols\" [(ngModel)]=\"selectedColumns\"
selectedItemsLabel=\"{0} Columnas Seleccionadas\" [style]=\"{'min-width': '200px'}\"
                        </p-multiSelect>
                        <span class=\"p-input-text\" type=\"text\"
                        (input)=\"dtable.filterGlobal($any($event.target.value), true)\"
placeholder=\"Buscar...\" />
                        </span>
                    </div>
                </div>
                <p-table #dtable [value]=\"selectedCabeceras\" [columns]=\"selectedColumns\"
pTemplate=\"header\" let-columns>
                    <tr>
                        <th style=\"width: 5rem; text-align: left; padding: 5px 10px 5px 0\">
                            <p-tableHeaderCheckbox></p-tableHeaderCheckbox>
                            </th>
                        <th style=\"width: 5rem; text-align: left; padding: 5px 10px 5px 0\">
                            <p-columnFilter type=\"text\" [field]=\"col.field\" display=\"menu\">
                                </p-columnFilter>
                            </th>
                        <th style=\"width: 5rem; text-align: left; padding: 5px 10px 5px 0\">
                            </th>
                    </tr>
                    <tr>
                        <td style=\"width: 5rem; text-align: left; padding: 5px 10px 5px 0\">
                            <p-tableCheckbox [value]=\"cabecera\"></p-tableCheckbox>
                            </td>
                        <td style=\"width: 5rem; text-align: left; padding: 5px 10px 5px 0\">
                            </td>
                        <td style=\"width: 5rem; text-align: left; padding: 5px 10px 5px 0\">
                            </td>
                    </tr>
                </table>
                </div>
            </div>
        </div>
    </div>
</div>

```

</td>

</tr>

style="display: flex; align-items: center;">

<p-button label="Recibos" (cli

button-primary mr-2"></p-button>

<p-button label="Pedidos" (click)="consult

primary"></p-button>

<div style="margin-left: 10px;">

</div>

</div>

</div>

</div>

</div>

</di

</ng-template>

<ng-template pTemplate="body" let-pedido

<td>{{ pedido.nomart }}</td>

<td>{{ ped

</td>

<td>{{ pedido.fecha }}</td>

<td>{{

<td>{{ pedido.tipoprecio }}</td>

<td>{{ pedido.tipodoc }}</td>

</p-table>

</div>

</div>

</div>

</div>

-->

<div class="car

< /spa

<td>{{ cabecera.emitado }}</td>

<td>{{ cabecera.saldo }}</td>

</tr>

<!-- Add this below the table to display the total sum -->

<div *ngIf="totalSum">

{{ totalSum }}</h5>

</div>

<hr>

<div cla

class="col-md-6 mt-5">

<div class="form-group row">

col-md-3">Fecha:</label>

<div *ngIf="this.letras_flag" class="col-m

class="ml-0" [(ngModel)]="FechaCalend" [showIcon]="true"

da

<label class="control-label text-right col-md-3 mt-1">OP:</label>

<select class="form-control custom-select" (change)="tipoDocCh

value="FC">FACTURA</option>

<option value="PR">PRESUPUESTO<

</div>

<label class="control-label text

class="col-md-9 mt-1 ">

<select class="form-control form-control

<option *ngFor="let vendedor of vendedores" [value]="vendedor.cod_ven">

</select>

</div>

class="col-md-6 mt-5">

<div class="form-group row">

label>

<div class="col-md-9 mt-1">

danger" [(ngModel)]="bonificacion"

(ngModelChange)="validat

<label class="control-label text-right col-md-3 mt-1">Interes:</la

<input type="number" class="form-control form-control-danger"

(ngModelChange)="validateValue(\$event)" placeholder="">

</div>

div>

<div class="mt-3 d-flex align-items-center">

<input type="text" class="form-control" placeholder="Buscar..." [(ngModel)]="searchText">

importe" class="mr-2">

<p-button label="Pagar" (click)="pago()" styleClass="btn btn-primary">

*ngIf="this.opcionesPagoFlag" class="btn-group ml-2">

<button type="button" class="btn btn-info dropdown-toggle" data-toggle="dropdown">

<button type="button" class="btn btn-info dropdown-toggle dropdown-toggle-split">

haspopup="true" aria-expanded="false">

</button>

<input type="text" class="form-control" placeholder="Buscar..." [(ngModel)]="searchText">

searchText">

</div>

</div>

</div>

</div>

components\calculoproducto\calculoproducto.component.html

```
<div class="row">
    <div class="col-lg-12">
        <div class="card">
            <div class="card-body">
                <div class="form-body">
                    <hr class="m-t-0 m-b-40">
                    <div class="col-md-6">
                        <div class="form-group row">
                            <div class="col-md-3">Nombre:</label>
                            <div class="col-md-9">
                                {{this.producto.nomart}} </p>
                            </div>
                        </div>
                    </div>
                    <div class="row">
                        <div class="form-group row">
                            <label class="control-label text-right col-md-3">Nombre:</label>
                            <div class="col-md-9">
                                <p class="form-control-static"> {{this.producto.nomart}} </p>
                            </div>
                        </div>
                        <div class="form-group row">
                            <label class="control-label text-right col-md-3">Stk Deposito:</label>
                            <div class="col-md-9">
                                <p class="form-control-static"> {{this.producto.exil}} </p>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

class="control-label text-right col-md-3">Stk C.Central:</label>

class="form-control-static"> {{this.producto.exi2}} </p>

div>

</div>

<div class="row">

class="form-group row">

<label class="control-label text-right col-md-3">Stk C.Central:</label>

<div class="col-md-9">

<p class="form-control-static">

</div>

</div>

class="form-group row">

<label class="control-label text-right col-md-3">Pcio Contado:</label>

<div class="col-md-9">

<p class="form-control-static">

</div>

</div>

<div class="col-md-6">

<div class="col-md-6">

class="control-label text-right col-md-3">Pcio Contado:</label>

class="form-control-static"> {{this.producto.precon}} </p>

</div>

<div class="col-md-6">

<div class="col-md-6">

class="control-label text-right col-md-3">Pcio Tarjeta:</label>

class="form-control-static"> {{this.producto.prefi2}} </p>

</div>

</div>

<div class="row">

class="form-group row">

<label class="control-label text-right col-md-3">Pcio Tarjeta:</label>

<div class="col-md-9">

```

<p class="form-control-static">

</div>

</div>

{{this.tipoPrecioString}}</h3>

<hr class="m-t-0 m-b-40">

md-6">

<div class="form-group row">

label>

<div class="col-md-9">

(ngModelChange)="calcularPrecioTotal($event)"

número">

</div>

</div>

md-6">

<div class="form-group row">

<div class="col-md-9">

</div>

</div>

div>

<div class="form-actions">

<div class="row">

<div class="row">

<div clas

<button type="submit" class="btn btn-success" (click)="comprar($event)"> <i

button>

</div>

</div>

md-6"> </div>

</div>

</div>

</form>

```

components\carrito\carrito.component.html

```
<div class="row">
  <div class="col-lg-12">
    <div class="card">
      <div class="card-body">
        <h6 class="card-subtitle">Lista de Productos</h6>
        <div class="table">
          <thead>
            <tr>
              <th>Nombre</th>
              <th>Precio</th>
              <th>Eliminar</th>
            </tr>
          </thead>
          <tbody>
            <tr *ngFor="let item of this.items">
              <td>
                <input class="sin-bordes" type="number" [(ngModel)]="item.cantidad"
                  (change)="calculoTotal()" min="1">
              </td>
              <td>{{item.precio * item.cantidad}}</td>
              <td><button class="btn btn-danger">Eliminar</button></td>
            </tr>
          </tbody>
        </table>
      </div>
      <div class="col-lg-6">
        <div class="card">
          <div class="card-body">
            <h3><strong>TOTAL: {{this.subtotal}}</strong></h3>
            <div class="col-lg-6 text-right">
              <button class="btn btn-warning" (click)="vaciarCarrito()">Vaciar Carrito</button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<hr>
<div class="row justify-content-center">
  <div class="form-group row">
    <div class="col-md-6">
      <label>Fecha de entrega</label>
      <div *ngIf="this.letras_flag" class="col-md-9">
        <input type="text" class="ml-0" [(ngModel)]="FechaCalend" [showIcon]="true" />
      </div>
    </div>
    <div class="col-md-3">
      <label class="control-label text-right col-md-3 mt-1">OP:</label>
    </div>
  </div>
</div>
```

```

<select class="form-control custom-select" (change)="tipoDocCh
value="FC">FACTURA</option>
<option value="ND">NOTA DEBITO<
CREDITO</option>
<option value="NV">DEVOLUCION</option>
<option value="CS">CONSULTA</option>

<label class="control-label text-right col-md-3 mt-1">Vendedor:</l

<select class="form-control form-control-danger" [(ngModel)]="
vendedor of vendedores" [value]="vendedor.cod_ven">
{{v

</div>
</div>
</div>

class="form-group row">
<label *ngIf="this.puntoVenta_flag"
md-3">P.Venta:</label>
<div *ngIf="this.puntoVenta_flag" class="col
control form-control-danger" [(ngModel)]="puntoventa"
(ngMo

</div>
<label *ngIf="this.letras_flag"
mt-1">Letra:</label>
<div *ngIf="this.letras_flag" class="col-md-9
control-danger" [(ngModel)]="letraValue">
<option *ngFor="l

</select>
</div>

label>
<div class="col-md-9 mt-1">
<input type="text" [(ngModel)]="numerocomprobante" (ngModelChange)="validateV
placeholder="" [disabled]="!inputOPFlag">
</div>

<div class="row mt-2 justify-content-end">
<div>

```

mr-3" (click)="finalizar()">Finalizar Venta</button>

</div>

components\condicionventa\condicionventa.component.html

```
<div class="card">
  <div class="card-body">
    <h4 class="card-title">{{this.clienteFrom

    <button type="button" class="btn btn-info">{{tipoVal}}</button>
    <button type="button" class="btn btn-info"
      toggle="dropdown"
      aria-haspopup="true" aria-expanded="false">
    </button>

    <input type="text" class="form-control" placeholder="Buscar..." [(ngModel)]="searchText">

    <a class="dropdown-item" (click)="selectTipo(item)">{{item.tarjeta }}</a>

    <p-table #dtable [value]="productos" [tableStyle]="{ 'min-width': '50rem' }" [pagination]
      [globalFilterFields]="['nomart', 'marca']">
      <ng-template pTemplate="caption">

      class="p-input-icon-left ml-auto">
      <input pInputText type="text"
      (input)="dtable.filterGlobal($any($event.target).value, 'contains')">

      <p-button icon="pi pi-file-excel" (click)="exportExcel()"
      button>
    </div>
  </ng-template>
  <ng-template pTemplate="caption">

  pSortableColumn="nomart" style="width:30%">Nomart<p-sortIcon field="nomart"></p-sortIcon>

  pSortableColumn="marca" style="width:15%">Marca<p-sortIcon field="marca"></p-sortIcon></th>
  style="width:10%">Precio<p-sortIcon
  field="precon"></p-sortIcon>

  pSortableColumn="prefi1" style="width:10%">Precio<p-sortIcon
  fi

  *ngIf="prefi2" pSortableColumn="prefi2" style="width:10%">Precio<p-sortIcon

  <th *ngIf="prefi3" pSortableColumn="prefi3" style="width:10%">Precio<p-sortIcon

  <th *ngIf="prefi4" pSortableColumn="prefi4" style="width:10%">Precio<p
  th>
```

```

        <th pSortableColumn="exi1" style="width:10%">Stk Dep<p-sortIcon field=
pSortableColumn="exi2" style="width:10%">Stk CC<p-sortIcon field="exi2"></p-sortIcon></th>
VV<p-sortIcon field="exi3"></p-sortIcon></th>
        <th pSortableColumn="exi4" style="width:10%">Stk CC<p-sortIcon field="exi4"></p-sortIcon></th>
        <th style="width:20%">Select</th>
        <th style="width:20%">Select</th>
    </thead>
    <ng-template pTemplate="body" let-producto>
        <tr>
            <td>{{ producto.marca }}</td>
            <td *ngIf="prefi0">{{ producto.precon }}</td>
            <td>
                <td *ngIf="prefi2">{{ producto.prefi2 }}</td>
                <td *ngIf="prefi4">{{ producto.prefi4 }}</td>
                <td>{{ producto.exi1 }}</td>
                <td>{{ producto.exi3 }}</td>
                <td>{{ producto.exi4 }}</td>
            </td>
            <td>
                button icon="pi pi-cart-plus" (click)="selectProducto(producto)"
            </td>
        </tr>
    </ng-template>

```


components\condicionventacab\condicionventacab.component.html

```
<p>condicionventacab works!</p>
```

components\cuentacorriente

```
<div class="row">
  <div class="col-12">
    <div class="card">
      <div class="card-body">
        <p-table #dtable [value]="clientes" [tableStyle]="{ 'min-width': '50rem' }" [pTableStyle]="{ 'min-width': '50rem' }" [pTableFilterFields]="[globalFilterFields]" [globalFilterFields]="['nombre', 'direccion', 'dni', 'cuit', 'tipoiva', 'vendedor', 'telefono']">
          <ng-template pTemplate="caption">
            <div class="p-d-f">
              auto">
                <input pInputText type="text"
                  'contains')"
                  placeholder="Buscar.." />
                excel" (click)="exportExcel()"
                  styleClass="p-button-sm p-button"
                  icon="pi pi-plus-circle" [routerLink]="['../newcliente']"
                  style="width: 50px; height: 30px; border: 1px solid #ccc; border-radius: 4px; text-align: center; line-height: 30px; font-size: 12px; font-weight: bold; color: #000; background-color: #fff; cursor: pointer; margin: 2px; display: inline-block; vertical-align: middle; white-space: nowrap; text-decoration: none; outline: none; transition: background-color 0.2s ease-in-out, color 0.2s ease-in-out, border-color 0.2s ease-in-out, border-radius 0.2s ease-in-out;"/>
            </div>
          </ng-template>
          <ng-template pTemplate="body">
            <table>
              <thead>
                <th pSortableColumn="nombre" style="width:20%">Nombre<p-sortIcon
                  <th pSortableColumn="direccion" style="width:20%">Direccion<p-sortIcon
                  <th pSortableColumn="dni" style="width:20%">Dni<p-sortIcon
                  <th pSortableColumn="cuit" style="width:20%">Cuit<p-sortIcon field="cuit"></p-sortIcon></th>
                  <th pSortableColumn="tipoiva" style="width:20%">Iva<p-sortIcon field="tipoiva"></p-sortIcon>
                  <th pSortableColumn="vendedor" style="width:20%">Vendedor<p-sortIcon
                  <th pSortableColumn="telefono" style="width:20%">Tel.<p-sortIcon
                  <th>Select</th>
                <th style="width: 5rem"></th>
              </thead>
              <tbody>
                <tr>
                  <td>{{ cliente.direccion }}</td>
                  <td>{{ cliente.dni }}</td>
                  <td>{{ cliente.cuit }}</td>
                  <td>{{ cliente.tipoiva }}</td>
                  <td>{{ cliente.vendedor }}</td>
                  <td>{{ cliente.telefono }}</td>
                  <td>{{ cliente.select }}</td>
                </tr>
              </tbody>
            </table>
          </ng-template>
        </p-table>
      </div>
    </div>
  </div>
</div>
```

<td>{{ cliente.tipoiva }}</td>

<td>{{ c

<td>{{ cliente.telefono }}</td>

<td>

styleClass="p-button-sm p-button-info"> </p-button>

</ng-template>

</p-table>

</div>

</div>

components\editcliente\editcliente.component.html

```
<div class="row">
  <div class="col-lg-12">
    <div class="card">
      <div class="card-body">
        [formGroup]="editarclienteForm" (ngSubmit)="guardar(editarclienteForm)"
        cla

        <h3 class="box-title">Edicion Cliente</h3>
        <hr>

        <div class="col-md-6">
          <div *ngIf="!this.nombreFlag" class="form-group">
            <label class="control-label text-right col-md-3">Nombre</label>
            <input type="text" class="form-control" formControlName="nombre" placeholder="Nombre" />
            <small class="form-control-feedback"> Solo letras y mas de 2 de longitud</small>
          </div>
          <div *ngIf="this.nombreFlag" class="form-group row">
            <div class="col-md-3">Nombre</label>
            <div class="col-md-9">
              <input type="text" class="form-control" formControlName="nombre" placeholder="Nombre" />
              <small class="form-control-feedback"> Solo letras y mas de 2 de longitud</small>
            </div>
          </div>
        </div>
        <div *ngIf="!this.direccionFlag" class="form-group">
          <label class="control-label text-right col-md-3">Direccion</label>
          <input type="text" class="form-control" formControlName="direccion" placeholder="Direccion" />
          <small class="form-control-feedback"> Letras, numeros y mas de 2 de longitud</small>
        </div>
        <div *ngIf="this.direccionFlag" class="form-group row has-danger">
          <div class="col-md-3">Direccion</label>
          <div class="col-md-9">
            <input type="text" class="form-control" formControlName="direccion" placeholder="Direccion" />
            <small class="form-control-feedback"> Letras, numeros y mas de 2 de longitud</small>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
</div>

</div>

<div class="col-md-6">

<div *ngIf="

<label class="control-label text-right col-md-3">CUIT</label>

<input type="number" class="form-control" formControlName="cuit" placeholder="">

<small class="form-control-feedback"> 11 numeros con este formato: 20332451264

div>

</div>

<div *ngIf="!this.cuitFlag" class="form-group row has-danger">

<label class="control-label text-right col-md-3">CUIT</label>

type="number" class="form-control" formControlName="cuit" placeholder="">

este formato: 20332451264

</small>

</div>

<div class="col-md-6">

<div *ngIf="!this.dniFlag" class="form-group row has-danger">

<label class="control-label text-right col-md-3">DNI</label>

type="number" class="form-control" formControlName="dni" placeholder="">

este formato: 31267853

</small>

</div>

</div>

<div class="col-md-6">

<div *ngIf="!this.telefonoFlag" class="form-group row has-danger">

<label class="control-label text-right col-md-3">Telefono</label>
```

<div class="col-md-9">

formControlName="telefono" class="form-control"

pla

control-feedback"> De 5 a 15 numeros de longitud </small>

<

<div *ngIf="!this.telefonoFlag" class="form-group row">

md-3">Telefono</label>

<div class="col-md-9">

formControlName="telefono" class="form-control"

pla

control-feedback"> De 5 a 15 numeros de longitud </small>

<

</div>

<div class="col-md-6">

<div c

class="control-label text-right col-md-3">Cod. Iva</label>

formControlName="tipoiva" class="form-control"

(cha

<option value="Excento">Excento</option>

value="Monotributo">Monotributo</option>

<option va

<option value="Responsable Inscripto">Responsable

</div>

</div>

<div class="col-md-6">

<div class="form-gro

text-right col-md-3">Ingresos Brutos</label>

<div class="col-md

formControlName="ingresos_br" class="form-control">

<op

<option value="si">Si</option>

</select>

</div>

</div>

<hr>

class="row">

<div class="col-md-6">

offset-3 col-md-9">

<button type="submit" class="btn btn-primary">

'guardar'">Guardar</button>

<button type="submit" class="btn btn-danger">

(click)="accion = 'eliminar'">Eliminar</button>

<button type="button" class="btn btn-link">

[routerLink]="['../puntoventa']">Cancel</button>

</div>

</div>

<div class="col-md-6"> </div>

</form>

</div>

</div>

</div>

</div>

```

<!-- ===== -->
<!-- Main wrapper - style
===== -->
<section id="wrapper" class="
images/motomatch.png);">
    <div class="login-box card">
        <div class="card-body">
            [formGroup]="forma" (ngSubmit)="ingresar(forma)"
            class="form-horizontal form-ma
            class="text-center db">
                    <input formControlName="email"
                        placeholder='Email'>
                </div>
            </div>

            class="col-xs-12">
                <input formControlName="password" name="password" class="form-control"
                placeholder="Contraseña (minimo 6 caracteres)">
            </div>

            class="col-xs-12">
                <select [(ngModel)]="sucursal" class="form-control">
                    <option value=2>Suc. Valle Viejo</option>
                    <option value=3>Suc. Guem
                </select>
            </div>

            class="col-xs-12">
                <button class="btn btn-info btn-lg btn-block text-up
                type="submit">Ingresar</button>
            </div>

            <input id="checkbox-signup" [(ngModel)]="check" name="check" type="checkbox"

            <label for="checkbox-signup"> Recordar login </label>
        </div>
    </div>
</div>
===== -->

```



```
<!-- End Wrapper -->  
<!-- ==
```

components\newcliente\newcliente.component.html

```
<div class="row">
    <div class="col-lg-12">
        <div class="card ">
            <div class="card-body">
                [formGroup]="nuevoclienteForm" (ngSubmit)="guardar(nuevoclienteForm)"
                class="form"
            </div>
            <h3 class="box-title">Nuevo Cliente</h3>
            <hr class="border-top-1px solid #ccc">
            <div class="col-md-6">
                <div *ngIf="!this.nombreFlag" class="form-group">
                    <label class="control-label text-right col-md-3">Nombre y Apellido</label>
                    <input type="text" class="form-control" formControlName="nombre" placeholder="">
                    <small class="form-control-feedback"> Solo letras y mas de 2 de longitud</small>
                </div>
                <div *ngIf="this.nombreFlag" class="form-group">
                    <label class="control-label text-right col-md-3">Nombre y Apellido</label>
                    <div class="col-md-9">
                        <div class="form-control" formControlName="nombre" placeholder="">
                            <small class="form-control-feedback"> Solo letras y mas de 2 de longitud
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-md-6">
                <div *ngIf="this.direccionFlag" class="form-group">
                    <label class="control-label text-right col-md-3">Direccion</label>
                    <div class="col-md-9">
                        <div class="form-control" formControlName="direccion" placeholder="">
                            <small class="form-control-feedback">Letras, numeros y mas de 2 de longitud
                        </div>
                    </div>
                </div>
                <div *ngIf="!this.direccionFlag" class="form-group">
                    <label class="control-label text-right col-md-3">Direccion</label>
                    <input type="text" class="form-control" formControlName="direccion" placeholder="">
                    <small class="form-control-feedback">Letras, numeros y mas de 2 de longitud
                </div>
            </div>
        </div>
    </div>
</div>
```

```
<div *ngIf="this.telefonoFlag" class="form-group row has-dange
```

```
right col-md-3">Telefono</label>
<div class="col-md-9">
formControlName="telefono" class="form-control"
control-feedback"> De 5 a 15 numeros de longitud </small>
<div *ngIf="!this.telefonoFlag" class="form-group row">
md-3">Telefono</label>
<div class="col-md-9">
formControlName="telefono" class="form-control"
control-feedback"> De 5 a 15 numeros de longitud </small>
</div>
<div class="col-md-6">
class="control-label text-right col-md-3">Cod. Iva</label>
formControlName="cod_iva" class="form-control"
<option selected="true" value="4">Excento</option>
value="3">Monotributo</option>
<option value="2">Co
<option value="1">Responsable Inscripto</option>
</div>
</div>
<div class="col-md-6">
class="control-label text-right col-md-3">Ingresos Brutos</label>
<select formControlName="ingresos_br" class="form-control">
<option value="si">Si</option>
</div>
</div>
class="form-group row">
```

```

<label class="control-label text-right">
    class="col-md-9">
        <select class="form-control" formControlName="vendedor">
            (change)="onSelectionVendedorChange($event)">
                <option value="">
                    {{vendedor.vendedor}}</option>
                </option>
            </div>
        </div>
    </div>
    <div class="form-actions">
        <div class="row">
            <div class="col-md-6">
                <button type="submit" class="btn btn-success">Guardar</button>
            </div>
            <div class="col-md-6">
                <button type="button" class="btn btn-danger"
                    [routerLink]="['../puntoventa']">Cancel</button>
            </div>
        </div>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>

```

```

<div id="main-wrapper">
    <app-header></app-header>
    <app-sidebar></app-sidebar>
    <!-- Page wrapper -->
    <!-- =====
===== -->
    <!-- Container fluid
===== -->
    <div class="container-
outlet ></router-outlet>
    <!-- =====
===== -->
    <app-footer></app-footer>
===== -->
    <!-- Container fluid
===== -->
    </div>
    <!-- =====
Page wrapper -->
    <!-- ===== -->

```

components\pedidos\pedidos.component.html

```
<!-- pedidos-dialog.component.html -->
    <p-table [value]="pedidos">
        <ng-template pTemplate="h">

            <th>Nomart</th>
                <th>Tipo Doc.</th>
                    <th>Tarjeta</th>
                        <th>Cant.</th>

        </ng-template>
        <ng-template pTemplate="body" let-pedido>
            <tr>
                <td>{{ pedido.nomart }}</td>
                <td>{{ pedido.tipodoc }}</td>
                <td>{{ pedido.tarjeta }}</td>
                <td>{{ pedido.precio }}</td>
                <!-- Add more columns as needed -->
            </tr>
        </ng-template>
```

components\productos\productos.component.html

```
<button (click)="openModal()">Abrir Modal</button>
```



```

<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-body">
                <p-table #dtable [value]="clientes" [tableStyle]="{ 'min-width': '50rem' }" [p-
[globalFilterFields]="['nombre', 'direccion', 'dni', 'cuit', 'tipoiva', 'vendedor', 'telefono']"
                <ng-template pTemplate="caption">
                    <div class="p-d-f
auto">
                    <input pInputText type="text"
'contains')"
                    placeholder="Buscar.." />
                    excel" (click)="exportExcel()"
                    styleClass="p-button-sm p-button
icon="pi pi-plus-circle" [routerLink]="['../newcliente']"
                    style
                </div>
                </ng-template>
                <ng-template
                    <th pSortableColumn="nombre" style="width:20%">Nombre<p-sortIcon
                    <th pSortableColumn="direccion" style="width:20%">Direccion<p-sort
sortIcon></th>
                    <th pSortableColumn="dni" style="width:20%">Dni<p-sc
pSortableColumn="cuit" style="width:20%">Cuit<p-sortIcon field="cuit"></p-sortIcon></th>
                    style="width:20%">Iva<p-sortIcon field="tipoiva"></p-sortIcon>
                    </th>
                    style="width:20%">Vendedor<p-sortIcon
                    field="vendedor"></p-
pSortableColumn="telefono" style="width:20%">Tel.<p-sortIcon
                <th>Select</th>
                <th>Edit</th>
                <th style=
            </ng-template>
            <ng-template pTemplate="body" let-cliente>
                <td>{{ cliente.direccion }}</td>

```

```
<td>{{ cliente.tipoiva }}</td>
<td>{{ cliente.telefono }}</td>
<td>
    <p-button
        styleClass="p-button-sm p-button-info"> </p-button>
        <p-button icon="pi pi-pencil" styleClass="p-button-sm p-button
(click)="editCliente(cliente)"> </p-button>
    </td>
</div>
</div>
</div>
</div>
<!-- This is data table -->
```

```
<div *ngIf="errorMessage" class="error-message">
    {{ errorMessage }}
</div>
<div *ngIf="!errorMessage">

    <tr>
        <th>Fecha</th>
        <th>Importe</th>
        <th>Saldo</th>
        <th>Bonifi

    </tr>

    <ng-template pTemplate="body" let-recibo>
        <tr>
            <td>{{ recibo.fecha }}</td>
            <td>{{ recibo.bonifica }}</td>
            <td>{{ recibo.interes }}</td>
        </tr>
    </ng>
```

components\venta\venta.component.html

<p>venta works!</p>

shared/footer/footer.component.html

```
<!-- ===== -->
                                <!-- footer -->
                                <!-- =====

class="footer ">
    <div class="row ">
        <div class="col-sm-6 col-lg-6 align-self-center" >

style="color:#000000;">© 2023 EIS - Motomatch </h3>
                                </div>
                                </div>
                                </footer>
                                <!-- =====

End footer -->
    <!-- ===== -->
```

shared\header\header.component.html

```
<!-- ===== -->
<!-- Topbar header - start -->
===== -->
<header class="topbar">
<!-- ===== -->
<!-- Logo -->
===== -->
<div class="navbar-header">
  
  <b>
    
  </b>
  <!-- ===== -->
  <!-- End of Topbar header -->
  <div class="navbar-collapse">
    <ul class="nav navbar-nav">
      <li class="nav-item">
        <a class="nav-link nav-toggler hidden-md-up waves-effect waves-dark"
          href="javascript:void(0)"><i class="ti-menu"></i></a>
      </li>
      <li class="nav-item">
        <a class="nav-link nav-toggler hidden-md-up waves-effect waves-dark"
          href="javascript:void(0)"><i class="ti-menu"></i></a>
      </li>
    </ul>
    <!-- Central Text -->
    <ul class="nav navbar-nav">
      <li>
        <span class="navbar-text" style="color:#ffffff; font-size: 20px;font-weight:bold">
        </li>
      </ul>
      <!-- Cart Icon -->
```

```

                <a class="nav-link waves-effect waves-dark" [routerLink]="['carrito']" id=
<i class="mdi mdi-cart"></i>
                <div class="notify">

class="point">{{this.cantidad}}</span>
                </div>
                </a>

===== -->
                </div>
            </nav>
        </he
```

```

<!-- ===== -->
<!-- Left Sidebar ===== -->
<aside class="left-sid

<!-- Sidebar navigation-->
<nav class="sidebar-nav">
<ul>

<a class="has-arrow waves-effect waves-dark" href="#" aria-expanded="f
span></a>
<ul aria-expanded="false" class="collapse">

routerLinkActive="active" (click)="salir()" [routerLink]="['login']">Logout</a></li>

</li>

<li class="nav-devider">

dark" href='#' aria-expanded="false"><i class="mdi mdi-account-star"></i><span class="hide-men
expanded="false" class="collapse">

<!-- <li><a routerLinkActive="ac

<li><a routerLinkActive="active" [routerLink]="['adminusuarios']">

routerLinkActive="active" [routerLink]="['dispositivos']">Analisis Dispositivos</a></li>
routerLinkActive="active" [routerLink]="['dbmanager']">DB Manager</a></li>
routerLinkActive="active" [routerLink]="['configuraciones']">Configuraciones</a></li>
routerLinkActive="active" [routerLink]="['customimagenes']">Imagenes</a></li> -->

</ul>

</li>

expanded="false"><i class="mdi mdi-basket-fill"></i><span class="hide-menu">CLIENTES</span></a>

<li><a routerLinkActive="active" [routerLink]="['puntoventa']">Ingre

</ul>

</li>

expanded="false"><i class="mdi mdi-clipboard-outline"></i><span class="hide-menu">CUENTA CORRI
class="collapse">

<li><a routerLinkActive="active" [routerLink]="

```



```

        </li>

        <li> <a class="has-
mdi-chart-areaspline"></i><span class="hide-menu">ANALISIS</span></a>
        <ul>

            <li><a routerLinkActive="active" [routerLink]='[" analisiscaja"]">Caj
        </ul>
        </li>

        expanded="false"><i class="mdi mdi-arrow-down-bold-circle"></i><span class="hide-menu">EXPORTA
        class="collapse">

        <!-- <li><a routerLinkActive="acti

        </ul>
        </li>

        expanded="false"><i class=" mdi mdi-alarm-multiple"></i><span class="hide-menu">ALARMAS</span>

        <!-- <li><a routerLinkActive="active" [routerLink]="['alarms']">Conf
routerLinkActive="active" [routerLink]="['managealarms']">Administrar</a></li> -->

        </ul>
        </li>

        class="nav-devider"></li>
        <li> <a class="has-arrow waves-effect waves-dark"
i><span class="hide-menu">ADMINISTRADOR </span></a>
        <ul aria-expanded="

        </ul>
        </li>

        <li class="nav-devic
waves-dark" href='#' aria-expanded="false"><i class="mdi mdi-account-edit"></i><span class="hi
expanded="false" class="collapse">

        CLIENTE</a></li>
        <li><a routerLinkActive="active" [routerLink]='["u
routerLinkActive="active" [routerLink]="['formempresa']">CONSULTAS EMPRESAS</a></li>
routerLinkActive="active" [routerLink]="['consultas']">CONSULTAS USUARIOS</a></li>

```

```

        </ul>
    </li>
    <li class="nav-device"
        <li> <a class="has-arrow waves-effect waves-dark" href="#" aria-expanded=
menu">ESTRUCTURA </span></a>
        <ul aria-expanded="false" class="collapse"
routerLinkActive="active" [routerLink]="['estructura']">GENERAL </a></li>

        </ul>
    </li>

expanded="false"><i class="mdi mdi-border-color"></i><span class="hide-menu">EDICION </span></a>

        <li><a routerLinkActive="active" [routerLink]="['edicion']">CLIENT
routerLinkActive="active" [routerLink]="['edicionrubro']">RUBRO </a></li>
routerLinkActive="active" [routerLink]="['edicionsubrubro']">SUBRUBRO </a></li>

        </ul>
    </li>

expanded="false"><i class="mdi mdi-file-chart"></i><span class="hide-menu">HOME</span></a>

        <li><a routerLinkActive="active" [routerLink]="['portadas']">PORTAD
routerLinkActive="active" [routerLink]="['especiales']">ESPECIALES</a></li>
routerLinkActive="active" [routerLink]="['promos']">PROMO1</a></li>

        </ul>
    </li>

class="mdi mdi-send"></i><span class="hide-menu">NOTIFICACIONES </span></a>

        <li><a routerLinkActive="active" [routerLink]="['notificaciones']">

        </ul>
    </li>

class="mdi mdi-treasure-chest"></i><span class="hide-menu">UTILIDADES </span></a>

```

```

        <li><a href="https://fontawesome.com/icons?d=gallery&p=1" target="
*ngIf="visibleButton"><a href="https://analytics.google.com/analytics/web/#/p255500501/reports
params=_u..nav%3Ddefault%26_u.comparisonOption%3Ddisabled%26_u.date00%3D20210420%26_u.date01%3
        <li><a href="https://www.google.com/maps/@-28.447175,-65.7439136,1
<li><a href="http://getemoji.com/" target="_blank">EMOJIS </a></li>
f7b3-5015f69e0cf1/ad612ee565b85703f50c779e9452d679" target="_blank">REFERENCIA TAMAÑO DE IMAGE
a>
        <li> <a class="has-arrow waves-effect waves-dark" aria-expanded="false"><
        <ul aria-expanded="false" class="collapse">
a></li>
        </ul>
        </li> -->

</ul>

</nav>
        <!-- End Sidebar navig

<!-- ===== -->
        <!-- End Left S
===== -->
```

components\analisiscaja\analisiscaja.component.css

```
.p-d-flex {
  display: flex;
}
.justify-content-between {
  justify-content: space-between;
}
```

```
auto;D
    }
```

components\cabeceras\cabeceras.component.css

```
.altura-boton {  
    height: 50px; /* Ajusta este valor según sea necesario */  
}  
.scroll_list {
```

components\carrito\carrito.component.css

```
.sin-bordes {  
    border: none;  
    outline: none;  
}  
  
.sin-bordes:focus {  
    outline: none;  
}  
  
.center  
  
center;  
}
```

components\condicionventa\condicionventa.component.css

```
.scroll_list {  
    max-height: 200px;  
    overflow-y: auto;  
}
```

components\login\login.component

```
/*
  Template Name: Admin pro Admin
  Author: Wrappixel
  Email: niravjoshi87@gmail.com
  File: scss
*/
/*Theme Colors*/
/*bootstrap Color*/
/*Light colors*/
/*Normal colors*/

niravjoshi87@gmail.com
File: scss
*/
/*Theme Colors*/
/*bootstrap Color*/
/*Light colors*/
/*Normal colors*/

password Page
*****
.login-register {
  background-size: cover;
  background-repeat: no-repeat;

  width: 100%;
  padding: 10% 0;
  position: fixed; }

  .login-box {
    width: 360px; /*or 400px*/

    width: 100%;
    left: 0px;
    right: 0px; }

    .login-box .social {
      display: block;

    .login-sidebar {
      padding: 0px;
      margin-top: 0px; }

      .login-sidebar .login-box {
```


routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

redirectUnauthorizedToLogin = () => redirectUnauthorizedTo(['login']);

import {
  './guards/auth.guard';
  import { AngularFireAuthGuard, hasCustomClaim, redirectUnauthorizedTo, redirectUnauthorizedToLogin } from '@angular/fire/auth';
} from './guards/auth.guard';
import { PagesComponent } from './components/pages.component';
import { LoginComponent } from './components/login.component';

import { PuntoventaComponent } from './components/puntoventa/puntoventa.component';
import { EditclienteComponent } from './components/editcliente/editcliente.component';
import { NewclienteComponent } from './components/newcliente/newcliente.component';
import { CondicionventaComponent } from './components/condicionventa/condicionventa.component';
import { ProductosComponent } from './components/productos/productos.component';
import { CalculoproductoComponent } from './components/calculoproducto/calculoproducto.component';
import { CarritoComponent } from './components/carrito/carrito.component';
import { VentaComponent } from './components/venta/venta.component';
import { CuentacorrenteComponent } from './components/cuentacorrente/cuentacorrente.component';
import { CabecerasComponent } from './components/cabeceras/cabeceras.component';
import { AnalisiscajaComponent } from './components/analisiscaja/analisiscaja.component';

import { LogginguardGuard } from './guards/logginguard.guard';
import { SuperGuard } from './guards/super.guard';

import { AdminGuard } from './guards/admin.guard';

const APP_ROUTES: Routes = [

  { path: 'login', component: LoginComponent },
  { path: 'puntoventa', component: PuntoventaComponent, data: { titulo: 'Punto de Venta' } },
  { path: 'editcliente', component: EditclienteComponent, data: { titulo: 'Editar Cliente' } },
  { path: 'newcliente', component: NewclienteComponent, data: { titulo: 'Nuevo Cliente' } },
  { path: 'condicionventa', component: CondicionventaComponent, data: { titulo: 'Condicion de Venta' } },
  { path: 'productos', component: ProductosComponent, data: { titulo: 'Productos' } },
  { path: 'calculoproducto', component: CalculoproductoComponent, data: { titulo: 'Calculo de Producto' } },
  { path: 'carrito', component: CarritoComponent, data: { titulo: 'Carrito' } },
  { path: 'venta', component: VentaComponent, data: { titulo: 'Venta' } },
  { path: 'cuentacorrente', component: CuentacorrenteComponent, data: { titulo: 'Cuenta Corriente' } },
  { path: 'cabeceras', component: CabecerasComponent, data: { titulo: 'CC' } },
  { path: 'analisiscaja', component: AnalisiscajaComponent, data: { titulo: 'Análisis de Caja' } },
];

@NgModule({
  imports: [RouterModule.forRoot(APP_ROUTES)],
  exports: [RouterModule]
})
export class AppRoutingModule {}
```

```

        //{ path: 'puntosmedicion', component: PuntosmedicionComponent, data:{titulo: "
'chart/:keygrupo/:keymedicion/:nombreggrupo/:nombrepunto', component: ChartComponent, data :{ti
{titulo:'Admin'} ,canActivate: [SuperGuard]}},
        //{ path: 'noway', component: NowayComponent, c
WorkingComponent, data :{titulo:'Working'}}},
        //{ path: 'alarms', component: AlarmsComponent,
ManagealarmsComponent, data :{titulo:'Manage Alarms'}}},
        //{ path: 'customimagenes', component
Imagenes'},canActivate: [SuperGuard]}},
        //{ path: 'configuraciones', component: Configuracione
        //{ path: 'exportar', component: ExportarComponent, data :{titulo:'Exportar'},canActivate: [
ChartanalisisComponent, data :{titulo:'Analisis'}}},
        //{ path: 'adminusuarios', component: Adm
[AdminGuard]}},
        //{ path: 'dbmanager', component: DbmanagerComponent, data :{titulo:'DB manage
DispositivosComponent, data :{titulo:'Dispositivos Manager'},canActivate: [SuperGuard]}
        //{
{titulo:'Cammesa'} },
        //{ path: 'cammesahora', component: CammesahoraComponent, data :{titul
AduusuariosComponent, data :{titulo:'Usuarios'}, canActivate: [AdminGuard] },
        //{ path: 'not
{titulo:'Notificaciones'}, canActivate: [AdminGuard] },
        //{ path: 'analisis/:cd_et/:distrib
Distribuidor'} },
        //{ path: 'eventos', component: EventosComponent, data :{titulo:'Eventos'}
{titulo:'Scada'} },

{ path: 'login', component: LoginComponent },
        { path: '**', pathMatch: 'full', redirectTo: 'log

];

@NgModule({
    imports: [RouterModule.forRoot(APP_ROUTES)],
    exports: [RouterModule]
})
export

```

app.component.ts

```
import { Component,OnInit } from '@angular/core';
                                declare function init_plugins();
                                declare var j

templateUrl: './app.component.html',
                                styleUrls: ['./app.component.css']
                                })
                                export class AppComponent

console.log("APP COMPONENTS");
                                init_plugins();
```

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { DatePipe } from '@angular/common';

import { DynamicDialogModule } from 'primeng/dynamicdialog';
import { DropdownModule } from 'primeng/dropdown';

import { HttpClientModule } from '@angular/common/http';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { LoginComponent } from './login/login.component';

import { initializeApp, provideFirebaseApp } from '@angular/fire/app';
import { environment } from '../src/environments/environment';

import { provideAuth, getAuth } from '@angular/fire/auth';
import { provideDatabase, getDatabase } from '@angular/fire/database';
import { provideMessaging, getMessaging } from '@angular/fire/messaging';
import { provideStorage, getStorage } from '@angular/fire/storage';

import { provideAuthGuard } from '@angular/fire/compat/auth-guard';
import { FIREBASE_OPTIONS } from '@angular/fire/compat';

import { PagesComponent } from './components/pages.component';
import { FooterComponent } from './shared/footer/footer.component';

import { HeaderComponent } from './shared/header/header.component';
import { SidebarComponent } from './shared/sidebar/sidebar.component';

import { PuntoventaComponent } from './components/puntoventa/puntoventa.component';

import { ButtonModule } from 'primeng/button';
import { TableModule } from 'primeng/table';
import { CalendarModule } from 'primeng/calendar';

import { MultiSelectModule } from 'primeng/multiselect';
import { EditclienteComponent } from './components/editcliente/editcliente.component';

import { NewclienteComponent } from './components/newcliente/newcliente.component';
import { CondicionventaComponent } from './components/condicionventa/condicionventa.component';
import { ProductosComponent } from './components/productos/productos.component';
import { CalculoproductoComponent } from './components/calculoproducto/calculoproducto.component';
import { CarritoComponent } from './components/carrito/carrito.component';
import { VentaComponent } from './components/venta/venta.component';
import { FilterPipe } from './pipes/filter.pipe';
import { AnalisispedidosComponent } from './components/analisispedidos/analisispedidos.component';

```

```

import { CuentacorrienteComponent } from './components/cuentacorrie
components/cabeceras/cabeceras.component';
import { CondicionventacabComponent } from './compon
{ AnalisiscajaComponent } from './components/analisiscaja/analisiscaja.component';
import { Ped
{ RecibosComponent } from './components/recibos/recibos.component';
//import { DataTablesModule

@NgModule({
  declarations: [
    AppComponent,
    LoginComponent,
    PagesComponent,
    FooterC

    PuntoventaComponent,
    EditclienteComponent,
    NewclienteComponent,
    CondicionventaCompone

    CarritoComponent,
    VentaComponent,
    FilterPipe,
    AnalisispedidosComponent,
    Cuentacorr

    AnalisiscajaComponent,
    PedidosComponent,
    RecibosComponent,

    ],

  imports: [
    MultiSelectModule,
    //DataTablesModule,
    FormsModule,
    ReactiveFormsModule,
    BrowserModule

    HttpClientModule,
    DynamicDialogModule,
    DropdownModule,
    provideFirebaseApp(() => initi

    provideDatabase(() => getDatabase()),
    provideFunctions(() => getFunctions()),
    provideMes

```

```
],
  providers: [ DatePipe, AngularFireAuthGuard, { provide: FIREBASE_OPTIONS, useValue: env
{ } }
```

components\analisiscaja\analisiscaja.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Cabecera } from '../../../interfaces/cabecera';

import { CargardataService } from '../../../services/cargardata.service';
import { Cliente } from '../../../interfaces/cliente';

from '@angular/router';
import * as FileSaver from 'file-saver';
import { Xlsx } from 'xlsx/xlsx';

'sweetalert2'; // Import SweetAlert2
import { CrudService } from 'src/app/services/crud.service';

from 'primeng/api';
import { FilterService } from 'primeng/api';
import { DialogService } from 'primeng/api';

pedidos.component';
import { RecibosComponent } from '../recibos/recibos.component';

interface C

analisiscaja',
  templateUrl: './analisiscaja.component.html',
  styleUrls: ['./analisiscaja.component.css'],

AnalisiscajaComponent implements OnInit {
  cols: Column[];
  _selectedColumns: Column[];

  sucursal: SelectItem[];
  option2: SelectItem[];
  option3: SelectItem[];
  option4: SelectItem[];

  Cliente[];
  totalSaldosSeleccionados: number = 0;

  constructor(
    private _filterService: FilterService, private _crud: CrudService, private activatedRoute: ActivatedRoute, private _cargardataService: CargardataService
  ) {}

  ngOnInit() {
    this._crud.get('clientes').subscribe((data: any) => {
      this.clientes = data;
    });

    this._crud.get('sucursales').subscribe((data: any) => {
      this.sucursales = data;
    });

    this._crud.get('opciones').subscribe((data: any) => {
      this.opciones = data;
    });

    this._crud.get('tipos').subscribe((data: any) => {
      this.tipos = data;
    });

    this._crud.get('letras').subscribe((data: any) => {
      this.letras = data;
    });

    this._crud.get('vencimientos').subscribe((data: any) => {
      this.vencimientos = data;
    });

    this._crud.get('emitidos').subscribe((data: any) => {
      this.emitidos = data;
    });

    this._crud.get('iva').subscribe((data: any) => {
      this.iva = data;
    });
  }
}
```

```

        { field: 'bonifica', header: 'Bonifica' },
    ];
    this._selectedColumns = this.cols;
    this.getClientes();
    this.sucursal = sucursal;

    Guemes', value: 3 },
    { label: 'Deposito', value: 4 }
    ];
    }
    get selectedColumns(): Column[] {
        // Restore original order
        this._selectedColumns = this.cols.filter((col) =>
            col.selected);
    }

    comprobante: string) {
        this.dialogService.open(PedidosComponent, {
            data: {
                sucursal: sucursal,
                comprobante: comprobante,
                cod_sucursal: sucursal,
                comprobante: comprobante
            },
            header: 'Recibos'
        });
    }

    = localStorage.getItem('sucursal');
    this._cargardata.clisucx(sucursal).pipe(take(1)).subscribe(
        resp.mensaje;
        }, (err) => { console.log(err); });
    }

    integrarNombreClienteCabecera() {
        this.cabeceras.forEach(cabecera => {
            const cliente = this.clientes.find(cli => cli.idcli === cabecera.idcli);
            cabecera.nombrecliente = cliente.nombre;
        });
    }

```



```

    });
  }
  ngOnInit(): void {
    console.log('Inicio de la aplicación');

    this.selectedSucursal = event.value;
    console.log(this.selectedSucursal);
    this._cargardatos();

    console.log(data);
    this.cabeceras = data.mensaje;
    this.integrarNombreClienteACabeceras();

    this.calcularTotalSaldosSeleccionados();
  }
  consultaRecibo() {
    this.showRecibosDialog();
  }
  consultaPedidos() {
    this.showPedidosDialog();
  }
  calcularTotalSaldosSeleccionados() {
    this.saldos = this.cabeceras
      .reduce((sum, cabecera) => sum + Number(cabecera.saldo), 0); // Suma los saldos
  }
}

```

```
import { Component } from '@angular/core';
```

@Component

```
  analisispedidos.component.html',
```

```
    styleUrls: ['./analisispedidos.component.css']
```

```
  })
```

```
export class
```

components\cabeceras\cabeceras.component.ts

```
import { Component } from '@angular/core';
import { Cabecera } from '../../interfaces/cabecera';

import { CargardataService } from '../../services/cargardata.service';

import { ActivatedRoute, Router, NavigationEnd } from '@angular/router';
import * as FileSaver from 'file-saver';

import { Observable } from 'rxjs';
import { ObservableOf } from 'rxjs/operators';
import Swal from 'sweetalert2'; // Import SweetAlert2
import { CrudService } from '../../services/crud.service';

import { MotomatchBotService } from 'src/app/services/motomatch-bot.service';

import { Text } from '@angular/compiler';
import { pdfMake, pdfFonts } from 'pdfmake';

cabeceras.component.html',
styleUrls: ['./cabeceras.component.css'])
export class CabecerasComponent {

  Cabecera;

  public clienteFromCuentaCorriente: any;
  public selectedCabeceras: Cabecera[] = [];

  null;
  public importe: number = null;
  public tipo: any[] = [];
  filteredTipo: any[] = [];

  '';
  public listaPrecio: string = '';
  public activaDatos: number;
  public interes: number = 0;

  Dni: 0,
  Numero: 0,
  Autorizacion: 0
};
public cheque = {
  Banco: '',
  Ncuenta:
```

```

ImporteCheque: 0,
    FechaCheque: null
    };
    searchText: string;
    public opcionesPagoFlag: boolean;

FechaCalend: any;
    public inputOPFlag: boolean = true;
    public puntoVenta_flag: boolean = true;

"C"];
    public letraValue: string = "A";
    public tipoDoc: string = "FC";
    public puntoventa: number;

private myRegex = new RegExp('^[0-9]+$');
    public numerocomprobante: string;
    public cliente: string;

fecha_recibo: any;
    public recibos: Recibo[] = [];
    public numerocomprobanterecibo: number;

any;
    public puntoventaSelectedFormCabecera: any;
    public cabecerasFiltered: any[] = [];

constructor(private bot: MotomatchBotService, private _crud: CrudService, private activatedRoute: ActivatedRoute, private _router: Router) {
    this.getNombreSucursal();
}
ngOnInit(): void {
    this.clienteFromCuentaCorriente = JSON.parse(this.clienteFromCuentaCorriente);
    console.log(this.clienteFromCuentaCorriente);

    localStorage.getItem('sucursal');
    this._cargardata.cabecerax(sucursal, this.clienteFromCuentaCorriente);

    console.log(resp);
    this.cabeceras = resp.mensaje;
    }, (err) => { console.log(err); });

this._cargardata.tarjcredito().pipe(take(1)).subscribe((resp: any) => {
    this.tipo = resp.tipo;

//-----
// get vendedores -----

```

```

//-----
//get clientes-----
JSON.parse(localStorage.getItem('datoscliente'));
//-----
sucursal-----
this.sucursal = localStorage.getItem('suc

//get usuario-----
this.usuario = localStorage.getI

//-----
}
getNombreSucursal() {
this.

'1') {
    this.sucursalNombre = 'Casa Central';
}
else if (this.sucursal == '2') {

(this.sucursal == '3') {
    this.sucursalNombre = 'Suc. Guemes';
}
else if (this.sucur

filterByDay() {
    const dayOfWeek = new Date().getDay(); // Domingo - 0, Lunes - 1, ..., Sáb

'd2', // Lunes
    2: 'd3', // Martes
    3: 'd4', // Miércoles
    4: 'd5', // Jueves

dayFieldMap[dayOfWeek];
    this.filteredTipo = this.tipo.filter(item => item[dayField] === '1'

this.selectedCabeceras = event.sort((a: any, b: any) => {
    const dateA = new Date(a.emitió

- dateB.getTime());
});
let selectedCabecerasIniciales = this.selectedCabeceras;
this.

console.log(this.selectedCabecerasIniciales);
this.letraSelectedFormCabecera = this.selecte

```

```

this.selectedCabecerasIniciales[0].puntoventa;
this.opcionesPagoFlag = this.evaluateEventTypes(
    event: any[]): boolean {
    // Recorre cada objeto en el array event
    for (let item of this.selectedCabecerasIniciales) {
        console.log(item.tipo);
        // Verifica si la propiedad tipo es diferente de "FC", "ND", "NC"
        if (item.tipo !== "FC" && item.tipo !== "ND" && item.tipo !== "NC") {
            propiedad tipo no coincide, devuelve falso
            this.tipoVal = "EFECTIVO";
            return false;
        }
        devuelve verdadero
        return true;
    }
    calculateTotalSum(selectedCabeceras: any[]) {
        this.totalSum = 0;
        parseFloat(cabecera.saldo.toString()), 0);
    }
    // New function to handle the payment
    pago() {
        Swal.fire({
            icon: 'error',
            title: 'Error',
            text: 'El saldo total es 0.',
            showCancelButton: true,
            confirmButtonText: 'Aceptar',
            cancelButtonText: 'Cancelar'
        });
        Swal.fire({
            icon: 'error',
            title: 'Error',
            text: 'Por favor, ingrese un monto válido.',
            showCancelButton: true,
            confirmButtonText: 'Aceptar',
            cancelButtonText: 'Cancelar'
        });
        (this.importe > this.totalSum) {
            Swal.fire({
                icon: 'error',
                title: 'Error',
                text: 'El importe no puede ser mayor al total.',
                showCancelButton: true,
                confirmButtonText: 'Aceptar',
                cancelButtonText: 'Cancelar'
            });
        });
        return;
    }
    if (this.tipoVal == 'Condicion de Venta') {
        Swal.fire({
            icon: 'error',
            title: 'Error',
            text: 'Debe seleccionar una condición de venta.',
            showCancelButton: true,
            confirmButtonText: 'Aceptar',
            cancelButtonText: 'Cancelar'
        });
    }
}

```

```

        return;
    }
    this.generarSalida();
}

completos los campos necesarios
    try {
        await this.getNumeroComprobanteCabecera();

= await this.ajuste(this.selectedCabeceras);
        // Extracted filtering logic into a separate function

this.idAso();
        const psucursal = await this.generacionPagoPsucursal();
        const cabeceras = await this.generacionCabeceras();

await this.generacionRecibo(this.selectedCabecerasIniciales);
        let pagoCC = {
            psucursal: psucursal, // aca tengo el objeto cabecera
            recibo: recibo // aca tengo el objeto recibo
        };

generar los datos de pago:', error);
    }
}
// Define the filtering function
async function filterCabeceras(cabeceras) {
    this.cabecerasFiltered = [];
    cabeceras.forEach(cabecera => {
        Number(cabecera.ival)) {
            this.cabecerasFiltered.push(cabecera);
        }
    });
}

console.log(pagoCC);
this._cargardata.pagoCabecera(this.sucursal, pagoCC).pipe(take(1)).subscribe(
    () => {
        this.generarReciboImpreso(pagoCC);
        Swal.fire({
            title: 'Operación exitosa',
            text: 'Se generó el recibo correctamente.',
            icon: 'success',
            confirmButtonText: 'Aceptar'
        });
        this.incrementarNumeroComprobanteRecibo();
    },
    error => {
        Swal.fire({
            title: 'Error',
            text: 'No se pudo generar el recibo.',
            icon: 'error',
            confirmButtonText: 'Aceptar'
        });
    }
);

```

```
icon

realizar el pago.',
    });
    }
    }, (err) => { console.log(err); });
    }
    async g

fechaFormateada = fecha.toLocaleDateString('es-ES', {
    day: '2-digit',
    month: '2-digi

let year = fecha.getFullYear();
let month = fecha.getMonth() + 1;
let formattedMonth

(this.cabecerasFiltered.length > 1) {
    numero_int = 99999999;
    this.numero_fac = 99999

1 ? this.cabecerasFiltered[0].numero_int : null;
    this.numero_fac = this.cabecerasFiltered

console.log(numero_int);
    console.log(this.numerocomprobantecabecera + 1);
    //zona de corn

this.cliente.idcli = 0;
    }
    if (this.cliente.cod_iva == "") {
        this.cliente.cod_

= 0;
    }
    if (this.codTarj == "") {
        this.codTarj = "0";
    }
    let cabecera = { // es

this.numerocomprobanterecibo, //this.numerocomprobantecabecera + 1,
    puntoventa: Number(thi

this.letraSelectedFormCabecera, //this.letraValue,
    numero_fac: this.numero_fac,
    atipo

Number(this.cliente.idcli),
    cod_sucursal: Number(this.sucursal),
    emitido: fechaForma
```



```

parseFloat((this.importe / 1.21).toFixed(2)),//parseFloat((this.suma/1.21).toFixed(2)),//this.
1.21).toFixed(2)),//parseFloat((this.suma - this.suma/1.21).toFixed(2)),
                                                    iva2: 0,
                                                    iv

        saldo: 0,//this.suma,
                                dorigen: false,
                                cod_condvta: this.codTarj,
                                cod_iva:

aca hay que ver si se agrega un campo para elegir el nombre del vendedor
                                                    anulado: false,

es el que se logea?
                                turno: 0,
                                pfiscal: `${year}${formattedMonth}`,
                                mperc: 0,

formatDate(this.FechaCalend, 'dd/MM/yy', 'en-US'),//this.FechaCalend,//fecha de cierre con ca

id_aso: 0,
    }
        console.log(cabecera);
                                return cabecera;
                                }
                                async getNumeroComprobanteCa

this._cargardata.lastIdnum(this.sucursal).pipe(take(1)).toPromise();
                                                    console.log('NUMERO SE

= Number(numero['mensaje']);
    }
        async getNumeroComprobanteRecibo() {
                                let numero = await th

console.log('NUMERO SECUENCIAL:' + numero);
                                this.numerocomprobanterecibo = numero;
                                }
                                async

this._crud.incrementarNumeroSecuencial('recibo', this.numerocomprobanterecibo + 1).then(() =>

this.numerocomprobanterecibo = 0;
                                });
                                }
                                async generacionPagoPsucursal() {
                                let date = n

1) + "-" + date.getDate();
                                let hora = date.getHours() + ":" + date.getMinutes() + ":" + dat

console.log(Number(this.numerocomprobantecabecera) + 1);
                                let codtarj: number = 0;
                                if (th

```

```

Number(this.codTarj);
    }
    let numero_int;
    if (this.cabecerasFiltered.length > 1) {

this.cabecerasFiltered.length === 1 ? this.cabecerasFiltered[0].numero_int : null;
    }
    cantidad: 0, //nada
    precio: 0, //nada
    idcli: this.clienteFromCuentaCorriente.idcli,
    fecha: fecha,
    hora: hora,
    tipoprecio: "", //nada
    cod_tar: codtarj, //this.c
necesario
    numerotar: this.tarjeta.Numero, //si es necesario
    cod_mov: 0,
    suc_dest
this.tarjeta.Autorizacion,
    dni_tar: this.tarjeta.Dni,
    banco: this.cheque.Banco,
nombre: this.cheque.Nombre,
    plaza: this.cheque.Plaza,
    importeimputar: this.cheque.Im
fechacheque: this.cheque.FechaCheque,
    emailop: emailOp,
    tipodoc: "RC",
    puntoven
numerocomprobante: numero_int,
    estado: "",
    id_num: this.numerocomprobantecabecera +
console.log(psucursall);
    return psucursall;
    }
    async generacionRecibo(selectedCabecerasIni
selectedCabecerasIniciales.forEach((cabeceraInicial, index) => {
    let importe = 0;
    le
this.currentSaldoArray[index]);

```

```

        if (this.currentSaldoArray[index] <= this.importe) {

else {
    importe = this.importe;
    saldo = this.currentSaldoArray[index] - this.importe;

this.importe -= importe;
    console.log("currentsaldo:" + this.currentSaldoArray);
    let

numero_fac_cabecera = 0;
    }
    else {
        numero_fac_cabecera = Number(this.numero_fa

= {
    recibo: this.numerocomprobanterecibo, //generado con el serial desde firebase

factura de la cabecera fijo para el array
    c_cuota: 0, //fijo
    fecha: this.fecha_re

importe, //Number(selectedCabecerasIniciales[index].saldo) - selectedCabeceras[index].saldo, //

usuario: this.usuario, //fijo para el array
    observacion: 0, //fijo
    cod_lugar: '1'

this.letraSelectedFormCabecera, //this.letraValue, //letra de la cabecera fijo para el array
this.puntoventa, // punto de venta de la cabecera fijo para el array
    recibo_asoc: cabecera

    recibo_saldo: saldo, //cabeceraActual.saldo, //---> el saldo para cada array
    cod

fec_proceso: this.FechaCalend, //this.FechaCalend, //fecha de cierre con caja puede ser otro dia

    id_fac: this.numerocomprobantecabecera + 1 // fijo para el array
    };
    if (recibo

console.log(this.recibos);
    return this.recibos;
    }
    calculoImporteRecibo(cabecera: any, rem

parseFloat(cabecera.saldo.toString());
    if (remainingImporte >= saldoCabecera) {
        importe

```

```

return importe;
    }
    ajuste(selectedCabeceras: any[]) {
        this.currentSaldoArray = [];
        // V

this.totalSum) {
    let remainingImporte = this.importe;
    console.log("remainig importe:

y ajustar sus saldos
        selectedCabeceras.forEach(cabecera => {
            if (remainingImporte

parseFloat(cabecera.saldo.toString());
            if (currentSaldo <= remainingImporte) {

            } else {
                cabecera.saldo = currentSaldo - remainingImporte;
                remaining

        console.log("currentsaldo:" + this.currentSaldoArray);
    });
    // Devolver el ar

console.log(selectedCabeceras);
    return selectedCabeceras;
    } else {
        // Si el impor

situación como se prefiera
        console.error('El importe es mayor o igual al totalSum, ajuste

this.cabecerasFiltered.forEach(cabecera => {
        cabecera.id_aso = this.numerocomprobanterecibo;

        cabecera.anumero_com = this.numerocomprobanterecibo;
        cabecera.atipo = "RC";
    });

this.totalSum) {
    this.importe = this.totalSum;
    }
    }
    selectTipo(item: any) {
        consol

seleccionado
        this.codTarj = item.cod_tarj;
        this.listaPrecio = item.listaprecio;
        this.

```

```

this.abrirFormularioTarj();
// aca se llama a la funcion que muestra los prefijos
}

// aca se llama a la funcion que muestra los prefijos
}
}
abrirFormularioTarj() {

`<input type="text" id="titular" class="swal2-input" placeholder="Titular">
<input type="number" id="numero" class="swal2-input" placeholder="Número Tarjeta"
input" placeholder="Autorización">`,
showCancelButton: true,
confirmButtonText: 'Aceptar'

const titular = (<HTMLInputElement>document.getElementById('titular')).value;
const numero = (<HTMLInputElement>document.getElementById('numero')).value;
(<HTMLInputElement>document.getElementById('autorizacion')).value;
if (!titular || !numero || !autorizacion) {
Swal.showValidationMessage(`Por favor rellene todos los campos`);
//return;
}

RegExp("[0-9]{8}$");
let reTitular = new RegExp("[a-zA-Z ]{1,40}$");
let reAutorizacion = new RegExp("[a-zA-Z ]{1,40}$");

reNumero.test(numero) {
Swal.showValidationMessage(`El número de la tarjeta no es válido`);

Swal.showValidationMessage(`El DNI no es válido`);
//return;
}

Swal.showValidationMessage(`El titular no es válido`);
//return;
}
if (reAutorizacion.test(autorizacion)) {
Swal.showValidationMessage(`La autorización no es válida`);
//return;
}

}).then((result) => {

```

```

        if (result.value) {
            this.tarjeta.Titular = result.value;

this.tarjeta.Numero = result.value.numero;
            this.tarjeta.Autorizacion = result.value.autorizacion;

        title: 'Ingrese los datos del Cheque',
        html:
            `<input type="text" id="banco"
id="ncuenta" class="swal2-input" placeholder="Nº Cuenta">
            <input type="number" id="ncheque"
type="text" id="nombre" class="swal2-input" placeholder="Nombre">
            <input type="text" id="plaza" class="swal2-input"
type="number" id="importeimputar" class="swal2-input" placeholder="Importe a Imputar">
placeholder="Importe del Cheque">
            <input type="text" id="fechacheque" class="swal2-input"
placeholder="Fecha del Cheque">

Cambiar el tipo de input a 'date' para activar el datepicker nativo
            document.getElementById('fecha').type = 'date';

showCancelButton: true,
        confirmButtonText: 'Aceptar',
        cancelButtonText: 'Cancelar',

        (<HTMLInputElement>document.getElementById('banco')).value;
            const ncuenta = (<HTMLInputElement>document.getElementById('ncuenta')).value;
        = (<HTMLInputElement>document.getElementById('ncheque')).value;
            const nombre = (<HTMLInputElement>document.getElementById('nombre')).value;
        = (<HTMLInputElement>document.getElementById('plaza')).value;
            const importeimputar = (<HTMLInputElement>document.getElementById('importeimputar')).value;
        const importecheque = (<HTMLInputElement>document.getElementById('importecheque')).value;
        = (<HTMLInputElement>document.getElementById('fechacheque')).value;

ncheque || !nombre || !plaza || !importeimputar || !importecheque || !fechacheque) {

        //return;
    }

    let reBanco = new RegExp("[a-zA-Z ]{1,40}$");
    let reNcheque = new RegExp("[0-9]{1,40}$");
    let reNombre = new RegExp("[a-zA-Z ]{1,40}$");
    let rePlaza = new RegExp("[a-zA-Z ]{1,40}$");
    let reImporteCheque = new RegExp("[0-9]{1,40}$");

```

```

{1,40}$");
        if (!reBanco.test(banco)) {
            Swal.showValidationMessage(`El nombre d

reNcuenta.test(ncuenta)) {
            Swal.showValidationMessage(`El numero de cuenta no es váli

        Swal.showValidationMessage(`El numero de cheque no es válido`);
        //return;

        Swal.showValidationMessage(`El nombre no es válido`);
        //return;
    }
    if (!

es válida`);
    //return;
}
    if (!reImporteImputar.test(importeimputar)) {
válido`);
    //return;
}
    if (!reImporteCheque.test(importecheque)) {
        //return;
    }
    return { banco, ncuenta, ncheque, nombre, plaza, importeim

    if (result.value) {
        this.cheque.Banco = result.value.banco;
        this.cheque.No

        result.value.ncheque;
        this.cheque.Nombre = result.value.nombre;
        this.cheque.Plaz

        result.value.importeimputar;
        this.cheque.ImporteCheque = result.value.importecheque;

        console.log('Cheque guardado:', this.cheque);
    }
    });
}
    pendientes() {
        let missingF

        missingFields.push('Fecha');
    }
    if (!this.vendedoresV) {
        missingFields

```

```

Swal.fire({
    icon: 'error',
    title: 'Error..',
    text: 'Faltan datos!',

false;
    }
    else {
        return true;
    }
}
tipoDocChange(event) {
    console.log(event.t

== "FC") {
    this.inputOPFlag = true;
    this.puntoVenta_flag = true;
    this.letras_fl

this.inputOPFlag = true;
    this.puntoVenta_flag = false;
    this.puntoventa = 0;
    thi

this.inputOPFlag = true;
    this.puntoVenta_flag = false;
    this.puntoventa = 0;
    thi

this.inputOPFlag = true;
    this.puntoVenta_flag = false;
    this.puntoventa = 0;
    thi

this.inputOPFlag = false;
    this.puntoVenta_flag = false;
    this.puntoventa = 0;
    th

this.inputOPFlag = false;
    this.puntoVenta_flag = false;
    this.puntoventa = 0;
    th

this._cargardata.vendedores().subscribe((res: any) => {

```



```

        this.vendedores = res.mensaje;

string): boolean {
    return this.myRegex.test(value);
}
exportExcel() {
    import('xlsx').t

xlsx.utils.json_to_sheet(this.cabeceras);
    const workbook = { Sheets: { data: worksheet },

xlsx.write(workbook, { bookType: 'xlsx', type: 'array' });
    this.saveAsExcelFile(excelBuff

string): void {
    let EXCEL_TYPE = 'application/vnd.openxmlformats-officedocument.spreadsheet

data: Blob = new Blob([buffer], {
    type: EXCEL_TYPE
});
    FileSaver.saveAs(data, fileN

generarReciboImpreso(pagoCC: any) {
    // Calcular la suma de los importes de todos los recibos

recibo.importe, 0);
    let cliente = JSON.parse(localStorage.getItem('datoscliente'));
    cons

this.numeroAPalabras(totalImporte);
    let fechaActual = new Date();
    let fechaFormateada =

console.log(fechaFormateada);
    const tableBody = pagoCC.cabeceras.map(item => {
    const b

parseFloat(item.ival);
    return [
        item.tipo,
        item.letra,
        item.anumero_

ival).toFixed(2),
        item.saldo
    ];
    });
    console.log(tableBody);
    // Definir el

background: {
    canvas: [
        {
            type: 'rect',

```

BWYIDAoMDAsKcWsdNhIQDQ4RDqsgsLEBYQERUFUVD48XGBYUGBIUFRT/2wBDAQMEBAUEBQkFBQkUDQoSfNBQUFBQUFBQUFBQFEBAXEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGVFVFWVlhZWmNkZWZnaGlqc3Rlndd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLTlTNcICQoL/8QAtREAAgECBAQBDACFBAAQAAQJ3AAECAXEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYkpOUlZaXmJmaoqOkpaanqQkmqsrO0tba3uLm6wsPExcbHyMnK0tPUlDbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAQdyFVPC3jwwhg8SeILddk+tsENYwJzWqozmVh74DCrK0h2ufZ3xL+LXhD4QaDjRjXjHxDZaBpycebeTBSx9FHUn2Ar4B+NX/AjB+3F8TJR77xXrUQKXWrajiFTnOUT8uCDtXHB2pk8YxX6L3A/g1LR80/A8cGePbux8rpyv08c2YrGNX/diH3hn+9S1Y9ANqLVnTQLe8uNlhJFzr2t2tSfzEH3lKs5JzS5OB+dftD42/LT+E/xA802uhat4G10r+zLadbiOGzgW32uDKcrjIPcd64L9tDwAJtJ8D2Guf2itoWlnvLq4GZZnWxKgn5VyOfR07+3NNCKnUbXhr56/41/OlrHhnx9Hqc9rqukeLJNQZkjkhumdJc4KkbcwBf8aP7c04/8xC1/wC/6/41/N9J4f8AEUUbefpXiGAlizSTW9wulePkOV+YDHPfkYrLubua2JWd760cKxRZJ5Y8KeAcNjA+y/4J22f7SXwwj8a6z471rQ7e4meKCz09BgbQBVbB3z04r6Ft/+CQfgulGbf4k+MYZVBAeOZVIBqiT77jmsVcpIrj1Vs07beH7RV1qPg7xdpS+G/iToyE3djgoswU4Z1VuVYHqOnIxB9cfjs80wfeA79akpiExTdwBxnn60+uL+LXxV8M/BfwXf+KvFggq98SfGV9LbfDrT7Pwdo7MyxXV5Gtgxduo745VSR2r5y1r9rH40eKpCb74peIpJWOUjjjPIjQMM5AU/h7Urov1Z/QM11Cv3wDr1hX2varqMzm91FTurnGXdbm+eUkdB949R7danmHyM/ohl74yBfC8Tyat4w0WRPvGa+jGMe2a8k8zf8ABQJ4BeC24a5Wwn+xxcGR17ibftFxm5tXhBJJP+nsP12MCubdghUaOUF5ZVPBB9e516DilzXGon62+Kf+Cwfwv0+cp4d8L+JvEsYH+uEHL4aaP4zsdNvNltNSTelpfDEi400ove9fjp+xb+yFqP7U3juP7ZHJaeBdJZXv8AUPL+WbDZWKJuhdu5HQGv3E0PQ7DwvotJb/gol+3JqvjXxXqXwz8DahcWPhfS5fsur6hauU0oT5wlurj1VByD3JBr57/ZF/ZRlv8Aag8fRaRYY03wpphE2q6sq8RKf+WKKerNzjpPoselcz+0p8LNe+Dvxm8VaD4hs7iOew7ku7W4mJPn27ylw6N0ZsHGRhLSodL0uzjCBUA3ykDl5G6sx6kn1rsOKanenVpsZhj2owPSlooAgaxtpJPma3iaT++UBP509reJs5iQ50T1RzU1FAFG40XTotP0uyttOsYRtjtrSIrog9AAMCtCk20vSgCOXjB7fyr4F/bz+B+qfCvxVpX7SPw0tUj8ReH5hJrNmgw11D0MhA6kDg/XPavv1ldfXlqnq2k2esaZc6fe26XNncxtFLDIMq6kYIIoGfBGjF8FhPAdxYwNqfgfxLaXxUF47dVmjlY52MByM024/4LFeDd7K0iaXe3BjuLJmBzsOMMAehJBA4x3rxyH/AIJs/H+beB4St7bcGBP2xPLHAYvP3jjj+tr7XWh9KeKp+Cy6tbzL4b+GNypX/BLN486hKrs2Gh6W7nte9JRC8d3rj716N4V/416+Nrp1/4SDxzpelRkYk6wC07DPX7wXtRd7DTR8BMqtsG923FWRgpcX+MxhyDxP411VfAel3SjyLW5ti95NF61dLeNgI5HfnNfc+PwNl/4Jo/Cb40alb6xewXHjPXbdvMiuNYx5ETdiktPIUg85ya+sFj8S79b4YIE5tEaNsoWUAcGvin9oz9lxb9+zDrCweKLJLjSZJdlNrtqpa215xhifuHbn5T36V+6+ufe3wf4Z2/2r4n0jTuG3xglZNN8EeGdQl6ZozisBD5cXJKl5D8qkE5+hr7x+BP/BI28uriz1X4sa3HHbK3mPoOkE7jnbKvL745xX6N+B7Xwlp+kxh74es9B8O6Xb6NpFouyC0tUCoo9eOpPgea36TjrTqokKKKkACKzsfX0paY2elAA0nGQNxrzH4v/tLfDb4Fwo3jPxTZ6TPJbOP70Phm18N+GJ4z481tcQyMm9dPgJwZ3GRz1C+4r8e44fE3xY8auwglTxb4o1WcNtVGuZ53LAKnk/KAcYxjioZSjc/Sf4GfwK8a6j4n0KdVOo6PqdlJA/LZ+aCRiAHbGSrexruPan/AASp+NXizT473V20fwqrbxHa3lw00yqxOfMVQB9Oa29c/wCCSww9cfaNklKBZU5y0bY+ZG/21PB+lderbq+Iv8Agmp8Hfi38DNF8XEGPH+kjsJ9d+0Jc6an2QTKHP3ghHRcdfevtvxt7vatDJjWvgf9urwb+1F8UPGSfh/4dRHTAPlfLlCaVqSWLxcSbfm807gwUdaO6199MMjduPdzgYnJq4H4a6t/wTd/afbzbm78I2+oSzc+J3w71muPE3wz1SztY2Km6t7Bp414yGLtUIMZH+Nf0PEHGKjmhW4RklRZyz1VlBH69aLF8zP5vvBvjzxH4AvhN4d1/UDahjiVv9b91/8A4KsazpOoWPh/4wxR6nYSMIV8TWKBW5iAVpkHBHqRyO9fYX7QX7B/wr+PdhNNC6Fb+H/EQU+RrGmRiKQNZjEs2+OZvDPiO08y2ZW+x61ENSf7CTnK9cN0yKmw9Gz+gTR9asfEGL2uo6bcxXlhdIJYLiFgySKRKEGtCvzE/4JNftHXS2pahPn+1J8SL34rftAeN/EF9Mzg6i9lErJwlvC2wBQOCFIzn+IsfSv1Y/4J7/staV8FfhTpvijVLBH8deIoReXl3Mh8y2jcZWEpktuzGBraMxlg7doxkdqhast6KXo7fTrTxSYNKOksGctAFFIBQA6iikpALRSClpgFFFFFABRRRRQAuuuuAFFFFFABRRRSYoAAUu+Flh8QP2ZdalJ4I11PQSt9a3GwFhg4ZQ3UAjtXlwy18o/wDBS74gWfgn9lnxDasuV2zWGSxtoS21nZm5I/AGkxo/MEBW88TfGC98c3dmGOpW7C0FEW6sUe4ddvYMONwHUGv17NJDluOopKkOk4z4tffJw/8E/A+oeLvFFylrotjgzyRpvYZOBe9rDng+a+1GGVr3jxHeC2jljTJHKIRskkleK/P/wAF/sf618PvH2jeJRt45fC4X+jXQxgj/tLE+ZT8y9j3cZGR/9apehSSsex/8Ff30n+JdLgWyYwWrPF0iELHirDPueuOlep/tWfEr/hb3wpfwf8P/il4A0e71LSlytav9R1hf8uAQyXybnccjPpXwr4L/YDuAIR0zxH4u8RQaRY6nCs9nHKCZp1KhhtQDPQ14uv/AAVP+ABxnxXfFIDIB5sJOP0/zmvlr9rf8AZn1T42fHbWnDX4v/AA6s9FvhuS32+YkizzY2kZLZzjkY9uOMUXYtD7z+Hv/BRL4H/ErXVZeHtM8Vi2lG8cRwLqEdWLl56KGYyYTxW78cv2lvhd+zv4kjT8OHulu0WOBNYDS7gwG1AzE7w3A9+tfSX7V/8AwT5+Kvx6+LS+JtKlZQrXTYNPtdPtRcPJ5wEScyNzjYt07Yo8wduh7CQJPk46nivgtUP+CfuuaTqUlqhHxn+HOMxtu5ia3uNaYoJkcqVL7s/730elddH/AMEl/jDdCK4h8UeHJIpYlAnhml2sh7g5ggqj8AWxjxBfYIJz9gkxwe03ftXzn+11+zDfeONf8G+Goviv4B0PSPCOHw9imja7qYikjuApEkxj3DrxgmVONL/4JR/FbWNBxJrVvPwSxXiea2muJfjikirSSOMk8DLEYH419R2t5HeQRTvSPbMoeOsh5VlIorYCD6Yr8TnW/wCCfes6KfTaZf8AXq+GttegLP4P+DvD1lqMeqXGn6bDbm7hfekmLQbtb+IY6H0pxdyT0Cvnb4vft6fCP4HeOrjw4q1ml6tdZgiIwwSOKd1VWBxyB7V9DNtAJSM8qd8cU27B6n6bFB/40eGfip4N8UeUebxt0iZ2jEjltYMDqaa8hIvYWX8df2j/A370Ph+01jxxav9n2130LE3SNDJJJ

wBlmz8SaL4w+LvgbVdHvytxaW+m6rGxjuB8rkgndyBz2zXjf7TnwV8VftLfFLUfEWofGr4Z20lxkw6Tpgl3cIIBjkjfgM3
s43ml2fjbWHSbzUgzQW8ELSVtXqxA6D3r59/4Jv/s2f8KR/wCEw8QXvjPw14v+2KluLzw/eC4hhVTuIdsnYeOg4xXiH7VX
wCCmXwM8Ua5p+kadr15PfX86W9vGbN1Du5wBk4HWvfPih8U9C+Dvgm+8V+J7hrLR7FQ08iruYD6CvyK8M/sL6n4T8eaVNd
bI+C3iv9oP4Gjwj4QvdNt7u7mt5pbq+dhEY1IY7dvXP5UXYSSRXS/8FUP2f2XK+I7lvpYSe3tSn/gqf8AFyT4ivsA4z/Z8
wDgmN8QfG0c0vhr4k+CNfS1LYLI+m6hLcCBiM4baxGT15pXY/dP0t+CP7Znw1/aElnUNN8G6lcXklhb/AGm5kmt2iSNc9y2
ErSNX8eeGvD+v+KlhsbfWbu9NvDDDk+YnmMd2459fpXg3hP/gmn4y8YzT2vhX4peAtdltgHlh0vUGmeP8A2mCsc+vXPnF
cwrjzblkIzuHRR71N/wAOj/jKpJXxJ4bzwdxlm25z9efoeK9S+Fv7CvjT9nn4b/FSfxB4w8M6VqPiDShpdlqVldNBaxB2G
DvWpru9ndYhJ9jdXvJ2+9LIXHU9eelfVnxO+JehfCHWtqfizxNdGz0XTk8y4mVC5AzjgDkl+ZP7I/7L+j/A342aL4x8W/
Fz4c6jpemRSBYtPlDJu2YU4LY4POetfUn7ZPibwx+0F8FbzwX4P+KngnTb7UJ0Es2pasiiYlOWVSrdelUholzQ0/wD4Kc
K3JxlGPXBopi0NP8A4LGeKxrHj/wJ4T2rKLGzuLlothz+8KjdnPJHl8D3NfEutfCfVdC+G/hHxvLZbdE8RXV1BZyrnZvhl
wACdF+If/BOfwj4VWUwa3Z28d9YX0sYZILh5Pm752ncR0rN6laH5weBfhLqPjzRfFWuWVqk6F4csW1DVRxmIi8scIoP953
YKtPA+gRSNC6tqNkmqagRzczPy78n7ueAvYV5N/wTD8D2msfEnx4LyeSK4n8NyadFJEU/ynlKq8nzHnjPBqirqlz5Viks
CjXY9G+IGg3mowaVJHawRT72dhHtUBeM1+MvxO+EtX8K/HOs+FZdQglJtPuWhF1503zV/hyvbPQgcYr118PCPLlbYAliC
+uuagmk6LqF5K6xQ28DybwzFCqT/Svzj/4Jo/sz2/g++b4t6/qf23UJY2g02yskOyBZFpmpIzHLMQcD0r67/as+JsHhf8A
Gv6GtPs4tI0u2tIkCQWsKxIo6KqqAAPbAr8Mv2FPBaat+1T4IivZFe2t7p7tvLBVm2ozBeD0JwcdOK/bqTxZZTQupjuAHU
wBpD4L+Afhl4a0OP4ieHlft9JhiMaXYJLiIZwMdzmvvg/as+BM3we+M3iDSDqkeo6fdXTahBJ5ZWVU1OcNzjcPUeprxqLw
vd/0pXdy3ZnWw28nxU+PW8IlxNrfiVtoJJ3qZuAf95R17Yr+hzS90h0fTrWxtty29tEsUYJyDqjA5r8lv+Cdf7LNV4i8aw
wAapEvcd4wlaLQfCur6lPzHa2skrfQKTX88Wg+FL342fFs6VpsZklfxJqMvksucMzuWHGe+316Zr9s/2xviNDof7M/
j64to5hcSabLDGxRcKWGMnmvy6/4J7+GYLz9qvWU12xkTT45b0quRghSFVeeRlsnPXFHDHGSr83+JPDjeG9e1fQ76zW3vDN
hyGCHVW0NoRmS+3BAxOcbOAxHc9a+If27tQuPGX7V3jMlgq2rR2MKyA4AVRt4B4AyentU2ZfMe8/Cfx5H+zj/
wAE19Vle0a0013xhfTWdJEUwVixMZZQcEAKGOfavkn9mHxx4R+Enxj0Txx4u0e81qx00NPFbPqhme45ALqx5GcnNdv+094
DXiTxVquuPrOqwm6kGmyCOFAzFQoBPPCjNGorqzPg/4h+IdM8U/FnXPE2mwTWGnaJrJvoYpY8PbxtKGUdeScc/n2xX9C3w
br4B+L7eb4M+CmmFxlL/AGVbo7MASSEUdS3Tjj2qokbo/07/AiLBeKhrHxU8IeHHKtFp+ny3OFyGDOy4OQfRT+tfSH/BJr
b+1qbx/8AtR+LblJpIo7NE01FkHICgk9D90kl+o37H8Nn4E/
Zt8C6TmaVlsVmZkGV3Od5Ay3Tmi2o5M+Yv+CxniiOPwf4G8Nkbhc3zXckZ6FUAHJzwOfSuD/4JX+PPH98J9H8a6x4r8V6F
wVW+E7eOvCeg+PtN1BbV9DkNtPaXUWRKknQqQeMY5B4NfmLL4fLSM8pgkLbseZHv59DntRfUFZ7n74n9r74LJhT8TPDw9j
AIKlfF7QfFn70Ph2Dw3rFvrGn6zqoH2i10+OREB38gjocV+W8nhzbCQY7OQ7SAzWy56dz/njFfTf7UVgdF+DfwL8I2JQWc
aksddufDd7pml22kOsTnUA/753GRjGcHbWV+05+yfrH7LWqaVpniHVtLle/1hDJELEMDGq4znJ4+vev0S/4JW2Nn4N+Auq
wD4KeeIpxGX7RbJARBBabyRwx+cgiZi5LbjjJGOLtBqHNqfSH/BI34c2cnwb8Vape2G37bqqiOTBXeqIQeMnuT3or2f/AIJ

margin: [0, 0, 80, 0], // izquierda, superior,derecha , inferior

},

{

{ text: 'Vicario Segura 587\n' },

{ text: 'Capital - Catamarca\n' },

'3834-4172012\n' },

{ text: 'motomatch01@gmail.com' },

],

{

text: [

{ canvas: [{ type: 'rect', x: 0, y: 0, w: 10

'\n', style: { fontSize: 40 }, margin: [10, 5, 0, 0] },//{ text: this.letraValue + '\n', style

'this.letra' represente la letra seleccionada en el campo 'letra'

{ text: 'RECIBO'

fontSize: 12,

},

{

text: [

{ text: 'RECIBO'

+ '\n', alignment: 'right' },

```

        { text: 'Punto de venta: ' + pagoCC.recibo[0].c_p
    },
    {
        text: 'Fecha',
        margin: [25, 0, 5, 30],
        fontSize: 10,
    },
    {
        canvas: [
            {
                x2: 380, y2: 0,
                lineWidth: 2,
                lineColor: '#cccccc' //
            },
            {
                margin: [0, 0, 30, 0] // Agregar un margen inferior a la línea
            },
            {
                col
            },
            {
                text: 'Recibimos de: ', bold: true },
                { text: cliente.nombre + '\n' },
                { text: 'DNI: ' + cliente.dni + '\n' },
                { text: 'CUIT: ' + cliente.cuit + '\n' },
                '\n' },
            ],
            fontSize: 10,
            margin: [0, 10, 0, 10],
        },
        type: 'line',
        x1: 0, y1: 0,
        x2: 380, y2: 0,
        para la línea
    },
    {
        margin: [0, 0, 30, 20] // Agregar un margen in
    }

```

```

widths: ['*'],
    body: [
        [
            {
                text: [
                    numeroenPlabras + '\n', },
                { text: '\n' },
                { text: 'Retenidos: ' + numeroenPlabras + '\n' },
                { text: '\n' },
                { text: 'Neto a Cobrar: ' + numeroenPlabras + '\n' },
            ],
            style: 'total',
            alignment: 'left',
        ],
    ],
    layout: {
        hLineWidth: function (i, node) {
            return (i === 0 || i === 1) ? 1 : 0;
        },
        vLineWidth: function (i, node) {
            return (i === 0 || i === 1) ? 1 : 0;
        },
        hLineColor: function (i, node) {
            return '#000000';
        },
        vLineColor: function (i, node) {
            return '#000000';
        },
    },
    // ... Aquí puedes seguir añadiendo más elementos al documento ...

    del Pago' + '\n',
    alignment: 'center',
    bold: true,
    table: {
        ['Tipo', 'Letra', 'Comprobante', 'Sucursal', 'Emitido', 'Importe', 'Saldo'],
        [
            {
                bold: true,
            },
            {
                margin: [0, 0, 0, 30],
            },
            {

```

```

        body: [
            ['TOTAL APLICADO: $' + totalImporte],
        ],

    0, 0, 30], // Añade un margen inferior
    },
    {
        text: 'Recibí conforme: \n
        margin: [0, 10, 0, 0],
    },
    ],
    styles: {
        header: {
            fo

    },
    tableExample: {
        margin: [0, 5, 0, 5],
        fontSize: 8,

    margin: [0, 10, 0, 0],
    },
    },
    defaultStyle: {
    },
    };
    Swal.close();

    '_RECIBO_' + fechaFormateada + '.pdf');
    pdfMake.createPdf(documentDefinition).getBlob((blob

    pagoCC.recibo[0].recibo + '_RECIBO_' + fechaFormateada + '.pdf');
    }, (error: any) => {

    const unidades = ['CERO', 'UNO', 'DOS', 'TRES', 'CUATRO', 'CINCO', 'SEIS', 'SIETE', 'OCHO',
    'CATORCE', 'QUINCE', 'DIECISÉIS', 'DIECISIETE', 'DIECIOCHO', 'DIECINUEVE'];
    const decenas = [
    'SETENTA', 'OCHENTA', 'NOVENTA'];
    const centenar = ['', 'CIEN', 'DOSCIENTOS', 'TRESCIENTOS',
    'OCHOCIENTOS', 'NOVECIENTOS'];
    if (num < 10) return unidades[num];
    if (num < 20) return

    return decenas[Math.floor(num / 10)];
    return decenas[Math.floor(num / 10)] + ' y ' + unid

```

```

return centenas[Math.floor(num / 100)];
return centenas[Math.floor(num / 100)] + ' ' + th

% 1000 === 0) return unidades[Math.floor(num / 1000)] + ' MIL';
return unidades[Math.floo

(num < 1000000) {
    if (num % 1000 === 0) return this.numeroAPalabras(Math.floor(num / 1000)
    ' MIL ' + this.numeroAPalabras(num % 1000);
    }
    if (num < 1000000000) {
        if (num % 1000000 === 0) return this.numeroAPalabras(Math.floor(num / 1000000))
        MILLÓN';
        return this.numeroAPalabras(Math.floor(num / 1000000)) + ' MILLÓN ' + this.numero

```

components\calculoproducto\calculoproducto.component.ts

```
import { Component, Inject, OnDestroy } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { DynamicDialogConfig } from 'primeng/dynamicdialog';
import { CarritoService } from 'src/app/services/carrito.service';

templateUrl: './calculoproducto.component.html',
styleUrls: ['./calculoproducto.component.css'];

clienteFrompuntoVenta: any;

public producto: any;
public cliente: any;

public codTarj: any;
public listaPrecio: any;

public pedido: any = {
  'idven': 0,
  'fecha': '',
  'hora': '',
  'tipoprecio': '',
  'cod_tar': 0,
  'titulartar': '',
  'nomart': '',
  'nautotar': 0,
  'dni_tar': 0,
  'banco': '',
  'ncuenta': 0,
  'ncheque': 0,
  'importecheque': 0,
  'fechacheque': '01/01/1900'
};

public precio: number;
public precioTotal: number;

any;
public tipoPrecioString: string = '';

constructor(private _carrito: CarritoService, private _router: Router, private _config: DynamicDialogConfig) {
```



```

console.log("constructor");

this.config.data.cliente;
this.tarjeta = this.config.data.tarjeta;
this.cheque = this.config.data.cheque;

this.codTarj = this.config.data.codTarj;
this.listaPrecio = this.config.data.listaPrecio;

console.log("producto:" + JSON.stringify(this.producto));
console.log("cliente:" + JSON.stringify(this.config.data.cliente));
JSON.stringify(this.tarjeta));
console.log("cheque:" + JSON.stringify(this.cheque));
console.log("codTarj:" + JSON.stringify(this.codTarj));
console.log("listaPrecio:" + JSON.stringify(this.listaPrecio));

//case dependiente de this.listaPrecio en caso de 0 this.producto.precon , si es 1 this.producto.precon, si es 2 this.producto.precon, si es 3 this.producto.precon, si es 4 this.producto.precon
switch (this.listaPrecio) {
    case "0":
        this.precio = this.producto.precon;
        break;
    case "1":
        this.tipoPrecioString = 'Precio de Tarjeta';
        this.precio = this.producto.precon;
        break;
    case "2":
        this.tipoPrecioString = 'Precio de Tarjeta';
        this.precio = this.producto.precon;
        break;
    case "3":
        this.tipoPrecioString = 'Precio de Tarjeta';
        this.precio = this.producto.precon;
        break;
    case "4":
        this.tipoPrecioString = 'Precio de Tarjeta';
        this.precio = this.producto.precon;
        break;
}

this.ref.close();
}
}

calcularPrecioTotal(newValue: number) {
    console.log(newValue);
    console.log(this.precioTotal);
}

comprar(event: Event) {

```

```
event.preventDefault();
```

```
th
```

```
this._carrito.agregarItemCarritoJson('carrito', this.pedido);
```

```
this.precioTotal = 0;
```

```
this
```

es para que se muestre en la factura , presupuesto, etc el nombre del cliente

```
localStorage.
```

```
new Date();
```

```
let fecha = date.getFullYear() + "-" + (date.getMonth() + 1) + "-" + date.getDa
```

```
date.getSeconds();
```

```
console.log("PRECIO:" + this.precio);
```

```
//completar los datos del objet
```

```
(this.producto.idart != undefined) {
```

```
this.pedido.idart = parseInt(this.producto.idart);
```

```
= this.producto.nomart;
```

```
}
```

```
this.pedido.cantidad = this.cantidad;
```

```
this.pedido.precio =
```

el carrito me va sumado

```
if (this.cliente.idcli != undefined) {
```

```
this.pedido.idcli = pa
```

```
undefined) {
```

```
this.pedido.idven = this.cliente.cod_ven;
```

```
}
```

```
if (this.cliente.fecha !=
```

```
(this.cliente.hora != undefined) {
```

```
this.pedido.hora = hora;
```

```
}
```

```
if (this.listaPrecio
```

```
if (this.codTarj != undefined) {
```

```
this.pedido.cod_tar = parseInt(this.codTarj);
```

```
}
```

```
if
```

```
this.tarjeta.Titular;
```

```
}
```

```
if (this.tarjeta.Numero != undefined) {
```

```
this.pedido.numerot
```

```
(this.tarjeta.Autorizacion != undefined) {
```

```
this.pedido.nautotar = parseInt(this.tarjeta.A
```

```

this.pedido.dni_tar = parseInt(this.tarjeta.Dni);
    }
    if (this.cheque.Banco != undefined)

(this.cheque.Ncuenta != undefined) {
    this.pedido.ncuenta = parseInt(this.cheque.Ncuenta);

this.pedido.ncheque = parseInt(this.cheque.Ncheque);
    }
    if (this.cheque.Nombre != undefined)

(this.cheque.Plaza != undefined) {
    this.pedido.plaza = this.cheque.Plaza;
    }
    if (this.cheque.ImporteImputar != undefined) {
    this.pedido.importe =
= parseInt(this.cheque.ImporteImputar);
    }
    if (this.cheque.ImporteCheque != undefined) {
    this.pedido.importecheque =
    console.log("FECHA CHEQUE:" + this.cheque.FechaCheque);
    if (this.cheque.FechaCheque != undefined) {
    this.pedido.fechacheque = this.cheque.FechaCheque;
    }
    else {
    this.pedido.fechacheque = this.cheque.FechaCheque;
    }
}

```

```

import { Component } from '@angular/core';
//agregar importacion de router para navegacion
import { Router } from '@angular/router';
import { CarritoService } from 'app/services/carrito.service';
import { SubirdataService } from 'src/app/services/subirdata.service';
import { CargardataService } from 'src/app/services/cargardata.service';
import Swal from 'sweetalert2';
import { first, take } from 'rxjs/operators';
import { AngularFireDatabase } from '@angular/fire/database';
import { MotomatchBotService } from 'src/app/services/motomatchbot.service';
import * as pdfFonts from 'pdfmake/build/vfs_fonts';
pdfMake.vfs = pdfFonts.pdfMake.vfs;
import { formatDate } from '@angular/common';

@Component({
  selector: 'carrito-component',
  templateUrl: './carrito.component.html',
  styleUrls: ['./carrito.component.css']
})
export class CarritoComponent {
  ref: DynamicDialogRef;

  public suma: number = 0;
  public tipoDoc: string = "FC";
  public numerocomprobante: string = "";

  public sucursal: string = "";
  public numero: string = "";

  private myRegex = new RegExp('^[0-9]+$');

  public vendedor: string = "A";
  public vendedorV: any;
  public cliente: any;

  public puntoVenta_flag: boolean = true;
  public letras_flag: boolean = true;

  private bot: MotomatchBotService;
  private _crud: CrudService;
  private _subirdata: SubirdataService;

  router: Router;

  constructor(
    private _carrito: CarritoService,
    private _subirdata: SubirdataService,
    private _cargardata: CargardataService,
    private _firebase: AngularFireDatabase,
    private _bot: MotomatchBotService,
    private _crud: CrudService,
    private _router: Router
  ) {
    this.FechaCalend = new Date();
    this.getItemsCarrito();
    this.calcularTotal();
  }

  this.usuario = localStorage.getItem('usernameOp')

```

```

        this.cliente = JSON.parse(localStorage.ge

{
    let items = localStorage.getItem('carrito');
        if (items) {
            this.itemsEnCarrito = J

this._cargardata.vendedores().subscribe((res: any) => {
        this.vendedores = res.mensaje;

this.sucursal = localStorage.getItem('sucursal');
        if (this.sucursal == '1') {
            this.suc

        this.sucursalNombre = 'Suc. Valle Viejo';
        }
        else if (this.sucursal == '3') {

'4') {
        this.sucursalNombre = 'Deposito';
        }
    }

        initLetraValue() {
            if (this.cliente

else if (this.cliente.cod_iva == 1)//excento
    {
        this.letraValue = "A";
    }
    else if

"A";
    }
    else {
        this.letraValue = "B";
    }
}

        tipoDocChange(event) {
            console.

(this.tipoDoc == "FC") {
        this.inputOPFlag = true;
        this.puntoVenta_flag = true;

```

```

this.inputOPFlag = true;
this.puntoVenta_flag = false;
this.puntoventa = 0;
thi

this.inputOPFlag = true;
this.puntoVenta_flag = false;
this.puntoventa = 0;
thi

this.inputOPFlag = true;
this.puntoVenta_flag = false;
this.puntoventa = 0;
thi

this.inputOPFlag = false;
this.puntoVenta_flag = false;
this.puntoventa = 0;
th

this.inputOPFlag = false;
this.puntoVenta_flag = false;
this.puntoventa = 0;
th

un sweet alert para confirmar la eliminacion
Swal.fire({
title: 'Estas seguro?',

showCancelButton: true,
confirmButtonColor: '#3085d6',
cancelButtonColor: '#d33',

(result.isConfirmed) {
Swal.fire(
'Eliminado!',
'El item fue eliminado'

this.itemsEnCarrito.indexOf(item);
this.itemsEnCarrito.splice(index, 1);
localSt

this._carrito.actualizarCarrito(); // es para refrescar el numero del carrito del header

```

```

this.suma = 0;
    for (let item of this.itemsEnCarrito) {
        this.suma += item.precio * item

    }

    (this.itemsEnCarrito.length > 0) { //hacer si
        if (this.pendientes()) {
            Swal.fire({

            });
            this.indiceTipoDoc = "";
            console.log('TIPO DOC:' + this.tipoDoc);

            (this.tipoDoc == undefined || this.tipoDoc == "" || this.puntoventa == undefined) //if (this.tipoDoc == undefined || this.numerocomprobante == "" || this.puntoventa == undefined)
            {
                title: 'Error..',
                text: 'Faltan datos!',
                footer: 'Completar

            }

            if (this.tipoDoc == "ND") {
                this.indiceTipoDoc = "notadebito";

            }
            else if (this.tipoDoc == "FC") {
                this.indiceTipoDoc = "factura

            numero.toString();
            }
            else if (this.tipoDoc == "NC") {
                this.indice

            this.numerocomprobante; //numero.toString();
            }
            else if (this.tipoDoc == "NV")

            this.numerocomprobante = this.numerocomprobante; //numero.toString();
            }

            "presupuesto";
            let numero = await this._crud.getNumeroSecuencial('presupuesto').pipe

            numero);
            this.numerocomprobante = numero.toString();
            }
            else if (t

```

```

        let numero = await this._crud.getNumeroSecuencial('consulta').pipe(take(1)).toPromise();

        this.numerocomprobante = numero.toString();
    }

    let emailOp = local

    this.itemsEnCarrito.map(obj => {
        return {
            ...obj,
            emailop: emailOp,

            this.puntoventa,
            numerocomprobante: this.numerocomprobante,
            estado:

            this.numerocomprobanteImpresion = this.numerocomprobante;
            localStorage.setItem('numerocomprobanteImpresion', this.numerocomprobanteImpresion);

            let sucursal = localStorage.getItem('sucursal');
            let exi = 0; // ESTO LO HAY QUE PONERLO EN EL ESTADO

            (sucursal == "2") {
                exi = 3;
            }
            else if (sucursal == "3") {
                exi = 1;
            }
            this._subirdata.editarStockArtSucxManagedPHP(result, exi);
            console.log(data);
        });
        this.agregarPedido(result, sucursal);
    }
}

'Error..',
    text: 'No hay items en el carrito!',
    footer: 'Agregue items al carrito'

year = fechasinformato.getFullYear();
let month = fechasinformato.getMonth() + 1;
let fecha = fechasinformato.getFullYear() + '-' + month + '-' + fechasinformato.getDate();

number = 0
if (this.tipoDoc != "FC") {
    numero_fac = 0;
}

```



```

    }
    else {
        numero_fac

this.getCodVta();
        let saldo = this.sumarCuentaCorriente();
        let cabecera = {
            tipo: t

this.puntoventa,
            letra: this.letraValue,
            numero_fac: numero_fac,
            atipo: this.ti

            cod_sucursal: this.sucursal,
            emitido: fecha,
            vencimiento: fecha,
            exento:

this.suma/1.21,
            ival: parseFloat((this.suma - this.suma / 1.21).toFixed(2)),
            iva2: 0

this.suma,
            dorigen: true,
            cod_condvta: codvent,
            cod_iva: this.cliente.cod_iva,

un campo para elegir el nombre del vendedor
            anulado: false,
            cuit: this.cliente.cuit,

            pfiscal: `${year}${formattedMonth}`,
            mperc: 0,
            imp_int: 0,
            fec_proceso: t

this.FechaCalend,//fecha de cierre con caja puede ser otro dia ?
            fec_ultpago: null,

console.log(cabecera);
        return cabecera;
    }
    sumarCuentaCorriente(): number {
        console.log

this.itemsEnCarrito) {
            console.log(item);
            if (item.cod_tar === 111) {
                acumula

campo 'valor' que queremos sumar
        }

```

```

        }
        return acumulado;
    }

    getCodVta() {
        if (
            firstCodTar = this.itemsEnCarrito[0].cod_tar;
            for (let item of this.itemsEnCarrito) {

            return firstCodTar;
        }

        agregarProductos() {
            window.history.back();
        }
        validateValue(va

agregarPedido(pedido: any, sucursal: any) {
    let fecha = new Date();
    let fechaFormateada

    '2-digit',
        year: 'numeric'
    });
    let cabecera = this.cabecera(fechaFormateada, fecha)
    sucursal).pipe(take(1)).subscribe((data: any) => {
        console.log(data.mensaje);
        this.i

    this.suma);
        //actualizar indices
        if (this.indiceTipoDoc != "") {
            this._crud.i

        parseInt(this.numerocomprobante) + 1).then(() => {
            console.log('Numero secuencial inc

            Swal.fire({
                icon: 'success',
                title: 'Pedido enviado',
                text: 'El ped

            sucursal ' + localStorage.getItem('sucursal')

            })
            this.itemsEnCarrito = [];
            loca

            this._carrito.actualizarCarrito(); // es para refrescar el numero del carrito del header

```

```

= [];
    if (this.tipoDoc == "FC") {
        if (!this.FechaCalend) {
            missingFields.push('Fecha Calendario');
        }
        missingFields.push('Vendedor');
        if (!this.puntoventa) {
            missingFields.push('Punto de venta');
        }
        missingFields.push('Numero de Comprobante');
        else if (this.tipoDoc == "NC" || this.tipoDoc == "NV") {
            missingFields.push('Número de comprobante');
        }
        if (this.numerocomprobante) {
            missingFields.push('Número de comprobante');
        }
    }
    else if (this.tipoDoc == "PR" || this.tipoDoc == "CS") {
        if (!this.vendedoresV) {
            missingFields.push('Vendedor');
        }
    }
    (missingFields.length > 0) {
        Swal.fire({
            icon: 'error',
            title: 'Error..',
            text: missingFields.join(', ')
        })
        return false;
    }
    else {
        return true;
    }
}

imprimir(items: any, numerocomprobante: string, fecha: any, total: any) {
    let cliente = JSON.stringify(items);
    if (this.tipoDoc == "FC") {
        titulo = "FACTURA";
    }
    else if (this.tipoDoc == "NV") {
        titulo = "DEVOLUCION";
    }
}

```

```
else if (this.tipoDoc == "ND") {
```

G3xglZNN8EeGdQ16ZozisBD5cXJK15D8qkE5+hr7x+BP/BI28uriz1X4sa3HHbK3mPoOkE7jnBKvL745xX6N+B7Xwlp+kx
h74es9B806Xb6NpFouyC0tUCoo9eOpPgea36TjrTqokKKKKACkZsfX0paY2e1AA0nGQNxrzH4v/tLfDb4Fwo3jPxTZ6TPJ
bOP70Phm18N+GJ4z481tcQyMm9dPgJwZ3GRz1C+4r8e44fE3xY8auwglTxb4o1WcNtVGUz53LAKnK/KAcYxjioZSjc/Sf4
GfwK8a6j4n0KdVoo6PqdlJA/lZ+aCRiAHbGSrexruPan/AASp+NXizT473V20fwqrbxHa3lw00yqxOfMVQB9Oa29c/wCCS
w9cfaNKLKBZU5y0bY+ZG/2lPB+lderbq+Iv8Agmp8Hfi38DNF8XEGPH+kjs9D+0Jc6an2gTKHP3ghHRCdfevtxc7vatDJj
Wvgf9urwb+1F8UPGSfh/4dRHTfAPlfLcaVqSWLxcSbfm807gwUdAo6199MMjlpuDzgyNJq4H4a6t/wTd/aFbzbm78I2+oS
Zc+J3w7imuPE3wz1Szty2Km6t7Bp414yGLIuMZH+Nf0PEHGKjmhW4RklrZyZ1VlBH69aLF8zP5vvBvjzxH4AvhN4dl/UdA
hjivvb9l/8A4KsazpOoWPh/4wxR6nYSMIV8TWKBW5iAVpkHBHqRyO9fYX7QX7B/wr+PdhNNc6Fb+H/EQU+RrGmRiKQNZje
s2+OZvDPiO08y2ZW+x6lENSf7CTnK9cN0yKmw9Gz+gTR9asfEGL2uo6bcxXlhdIJYLiFgySKRkEGTcvzE/4JNftHXS2pa
Pn+1J8SL34rftAeN/EF9Mzg6i9lErJwlvC2wBQOCFIzn+IsfSv1Y/4J7/staV8FfhTpviJVLBH8deIoReXl3Mh8y2jcZWE
pktuzGBraMxlxg7douxkdqhast6Kxo7fTrTxSYNKOkgSgCtAFFIBQA6iikpALRSClpgFFFFFABRRRRQAUUUUAFfffABRRRSYoAW
AAUu+Flh8QP2ZdalJ4I1lPQSt9a3GwFhg4ZQ3UAjtXlwy18o/wDBS74gWfgn9lnxDasuv2zWGSxtoS2lnZm5I/AGkxo/ME
BW88TfGC98c3dmG0Pw7C0EFw6sUe4ddvyMONwHUGv17NJDluOopKKok4z4tfFjw/8E/A+oeLvFFylrotjgzyRpvYZOBhe9
Dng+a+1GGVrjxHeC2jljTJKISRkkleK/P/wAF/sf618PvH2jeJrT45fC4X+jXqXgj/tLE+ZT8yY3cZGR/9apehSSsex/8F
f3On+JdLgWyYwWrPF0iELHirDPueu0lep/tWfEr/hb3wpfwf8P/il4A0e7lSLytav9R1hf8uAqNyxYbnccjPpXwr4L/YDU
AIR0zxH4u8RQaRY6nCs9nHKCZp1KhhtQDPQ14uv/AAVP+ABxnxFfDIB5sJOP0/zmv1r9rf8AZn1T42fHbWnDX4v/AA6s9P
FvhuS32+YkizzY2kZLZzjkY9uOMUXYTd7z+Hv/BRL4H/ErXVZeHtM8Vi2lG8cRwLqEDwLI56KGYYyTxW78cv2lvhd+zv4k
jT8OHulu0WOBNYDS7gwG1AzE7w3A9+tfSX7V/8AwT5+Kvx6+LS+JtKlZQrXTYNptdPtRcPJ5wEScyNzjJYt07Yo8wduh7C
QJPk46nivgTUP+CfuuaTqUlhgHxn+HOMxtu5ia3uNaYOjkcqVL7s/730elddH/AMEl/jDdCK4h8UeHJIpYlAnhml2sh7g5
gqj8AWxjxBfYIJZ9gkxwe03ftXzn+1l+zDfeONf8G+Goviv4B0PSPCOhW9imja7qYikjuApEkxj3DrxgmVONL/4JR/FbWm
AbxJrVvpsXiea2muJFjikurSSOMk8DLEYH419R2t5HeQRTwSpPBMOeOSM5V1IyCD6Yr8TNW/wCCfes6XfTaZf8Axq+Gtte
gL4P+DvD1lqMeqXGn6bDbm7hfeKmlQBtb+IY6H0pxdyT0Cvnb4vft6fCP4HeOrjwJ4q1m6tdZgiWWSOK0dlVWBxyB7V9DN
AJSM8gd8cU27B6n6bfB/40eGfjp4Ng8UeEbxtQ0iZ2jEjLtYMDggg8jHvWX8df2j/A370Ph+01jxxqv9n2130Le3SNDJJI
wBlmz8SaL4w+LvgbVdHvytxaW+m6rGxjuB8rkngndyBz2zXjf7TnwV8VftLffLUfEwofGr4Z20lxkw6Tpgl3cIIBjkjfgM3
s43ml2fjbWHSbzUgzQW8ELSVtXqx6D3r59/4Jv/s2f8KR/wCEw8QXvjPw14v+2KluLzw/eC4hhVtUIdsnYeOg4xXiH7VX
wCCmXwM8Ua5p+kadr15PfX86W9vGbn1Du5wBk4HWvfPih8U9C+Dvgm+8V+J7hrLR7FQ08iruYD6CvyK8M/sL6n4T8eaVNd
bI+C3iv9oP4GjwJ4QvdtN7u7mt5pbq+dhEY1IY7dvXP5UXYSSRXS/8FUP2f2XK+I71vpYSe3tSn/gqf8AFyT4ivsA4z/Z8
wDgmN8QfG0c0vhr4k+CNfS1YLI+m6hLcCBiM4baxGT15pXY/dP0t+CP7Znw1/aElNUNN8G6lcXklhb/AGm5kmt2iSNc9y2
ErSNX8eeGvD+v+KlhsbfWbu9NvDDDK+YnmMd2459fpXg3hP/gmn4y8YzT2vhX4peAtldtgHlh0vUGmeP8A2mCsc+vXPNE
cwrjzblkIzuHRR71N/wAOj/jKpJXxJ4bzwDXlm25z9efoeK9S+Fv7Cvjt9nn4b/FSfXB4w8M6VqPiDShpdlqVldNBaxB2G
DvWpru9ndYhJ9jdXvJ2+9LixHU9eelfVnxO+JehfCHwTqfizzNdGz0XTk8y4mVC5AzjgDk1+ZP7I/7L+j/A342aL4x8W/
Fz4c6jpemRSBYtPlDDJu2YU4LY4P0etfUn7ZPibwx+0F8Fbzx4P+KngnTb7UJ0Es2pasiIYL0WVSrdelUh01zQ0/wD4Kd
K3Jx1GPXBopi0NP8A4LGeKxrHj/wJ4T2rKLGzuL1othz+8KjdnPJHl8D3NfEutfCfVdC+G/hHxvLzbdE8RXV1BZyrnZvhl
wACdF+If/BOfwj4VWUwa3Z28d9YX0sYZILh5Pm752ncR0rN6laH5weBfhLqPjzRfFWuWVqk6F4csW1DVRxmIi8scIoP953
YKtPA+gRSNC6tqNkmqagRzczPy78n7ueAvYV5N/wTD8D2msfEnx4LyeSK4n8NyadFJEU/ynlKq8nzHnjPBqirqlz5Viks
CjXY9G+IGg3mowaVJHawRT72dhHtUBeM1+MvxO+EtX8K/HOS+FZdQg1JtPuWhF1503zV/hyvbPQgcYr118PCPLlbYALiC
+uuagmk6LqF5K6xQ28DybzWFCqT/Svzj/4Jo/sz2/g++b4t6/qf23UJY2g02yskOyBZFPmPiZHLMQcD0r67/as+JsHhf8A
Gv6GtPs4tI0u2tIkCQWsKxIo6KqqAAPbAr8Mv2FPBaat+1T4IivZFe2t7p7tvLBV6m2ozBeD0JwcdOK/bqTxZZTQupjuAHU
wBpD4L+AfhL4a00P4ieHlft9JhiMaXYJLiIzWmdzmvyg/as+BM3we+M3iDSdqkeo6fdXtahBJ5ZWVU0cNzjcPUeprxqLw
vd/0pXdy3ZnWw28nxU+PW8I1xNrfiVtoJJ3qZuAf95R17Yr+hzS90h0fTrWxtty29tEsUYJyjqjA5r8lv+Cdf7Lnv4i8aw
wAapEvcd4wlaLQfCur6lPzHa2skrfQKTX88Wg+FL342fFs6VpsZklfxJqMvksucMzuWHGe+316Zr9s/2xviNDof7M/
j64to5hcSabLDGxRcKWGMnmvy6/4J7+GYLz9qvWU12xkTT45b0quRghSFveeRlsnPXFHDHGS83+JPDjeG9elfQ76zW3vdN
hyGCHVW0NoRmS+3BAxOcbOAxHc9a+If27tQuPGX7V3jMlgq2rR2MKyA4AVRt4B4AyentU2ZfMe8/Cfx5H+zj/
wAE19V1e0a0013xhfTwdjEuWVixMZZQcEAKGOfavkn9mHxx4R+Enxj0Txx4u0e81qx00NPFbpghme45ALqx5GcnNdv+094
DXiTxVquuPrOqwm6kGmyCOFAzFQoBPPCjNGorqzPg/4h+IdM8U/FnXPE2mwTWGnaJrJvoYpY8PbxtKGUdeScc/n2xx9C3w
br4B+L7eb4M+CmmFxxLL/AGVbo7MASSEUDS3Tjj2qokbo/07/AILBeKhrHxU8IEHHKtFp+ny3OFyGDOy4QqfRT+tfSH/BJR
b+1qbx/8AtR+LblJpIo7NE01FkHICgk9D90k1+o37H8Nn4E/
Zt8C6TmaVlsVmZkGV3Od5Ay3Tmi2o5M+Yv+CxniiOPwf4G8Nkbhc3zXckZ6FUAHJzwOfSuD/4JX+PPh98J9H8a6x4r8V6F
wVW+E7eOvCeg+PtN1BbV9DkNtPaXUWRKknQqQeMY5B4NfmLL4fLSM8pgkLbseZHV59DntRfUFZ7n74n9r74LJhT8TPDw9j
AIK1fF7QfFn7OPh2Dw3rFvrGn6zqoH2i1O+OREB38gjocV+W8nhzbCQY7OQ7SAZWy56dz/njFfTf7UVgdF+DfwL8I2JQWd
aksddufDd7pml22kOsTnUA/753GRjGcHbWV+05+yfrH7LWqaVpniHVTl1e/1hDJELEMDGq4znJ4+vev0S/4JW2Nn4N+Auq
wD4KeeIpvGX7RbJARBBabyRwx+cgZi5LbjjJGOLtBqHNqfSH/BI34c2cnwb8Vape2G37bqqiOTBXeqIQeMnuT3or2f/AIJ

margin: [0, 0, 80, 0], // izquierda, superior,derecha , inferior

},

{

{ text: 'Vicario Segura 587\n' },

{ text: 'Capital - Catamarca\n' },

```

'3834-4172012\n' },

        { text: 'motomatch01@gmail.com' },

    ],

    {
        text: [
            { canvas: [{ type: 'rect', x: 0, y: 0, w: 100, h: 100 }],
              this.letraValue + '\n', style: { fontSize: 40 }, margin: [10, 5, 0, 0] },//{ text: this.letraValue + '\n', style: {
Asegúrate de que 'this.letra' represente la letra seleccionada en el campo 'letra'

            { text: 'COMO FACTURA' }

        ],

        alignment: 'center',

    { text: titulo + '\n' },

        { text: 'Nº 0000 -' + numerocomprobante + '\n' },

    ' + this.puntoventa + '\n' },

    ],

    alignment: 'right',

    font: 'serif',

    'Fecha: ' + fecha,

    alignment: 'right',

    margin: [25, 0, 5, 30],

    font: 'serif',

    {
        type: 'line',

        x1: 0, y1: 0,

        x2: 380, y2: 30,

        Color gris claro para la línea

    }

    ],

    margin: [0, 0, 30, 0] // Agregar margen a la línea

    {
        text: [
            { text: 'Sres: ' + cliente.nombre + '\n' },

            { text: 'DNI: ' + cliente.dni + '\n' },

            { text: 'CUIT: ' + cliente.cuit + '\n' },

```

```

cliente.tipoiva + '\n' },
                                ],
                                fontSize: 10,
                                margin: [0,

canvas: [
                                {
                                    type: 'line',
                                    x1: 0, y1: 0,
                                    x2:

'#cccccc' // Color gris claro para la línea
                                }
                                ],
                                margin: [0, 0, 3

Aquí puedes seguir añadiendo más elementos al documento ...
                                {
                                    style: 'tableExa

'15%'],
                                body: [
                                    ['Cant./Lts.', 'DETALLE', 'P.Unitario', 'Total'],

                                ],
                                {
                                    style: 'tableExample',
                                    table: {
                                        widths: ['*']

                                // ... Añade más filas según sea necesario ...
                                },
                                bold: tr

{
    header: {
        fontSize: 10,
        bold: true,
        margin: [2, 0, 0, 10],

        fontSize: 8,
    },
    total: {
        bold: true,

```

fontSize: 8,

```
    },  
    };  
    // cerrar el loading  
    Swal.close();  
    // Crear el PDF  
    pdfMake.createPdf(  
        '_' + fechaFormateada + '.pdf');  
    pdfMake.createPdf(documentDefinition).getBlob((blob) => {  
        this.numerocomprobanteImpresion + '_' + titulo + '_' + fechaFormateada + '.pdf');  
    }, (error)
```


components\condicionventa\condicionventa.component.ts

```
import { Component } from '@angular/core';
import { CargardataService } from '../../../services/cargardataService';

import Swal from 'sweetalert2';
import * as FileSaver from 'file-saver';
import { saveAs } from 'file-saver';
import { saveAsExcelFile } from 'xlsx';

import { Router } from '@angular/router';
import { FilterPipe } from 'src/app/pipes/filter.pipe';
import { filter } from 'rxjs/operators';

import { CalculoProductoComponent } from '../calculoproducto/calculoProductoComponent';
import { DynamicDialogRef } from 'primeng/dynamicdialog';
import { ChangeDetectorRef } from '@angular/core';

@Component({
  selector: 'condicionventa',
  templateUrl: './condicionventa.component.html',
  styleUrls: ['./condicionventa.component.css']
})
export class CondicionventaComponent {
  public tipoVal: string = 'Condicion de Venta';
  public codTarj: string = '';

  public tipo: any[] = [];
  searchText: string;
  ref: DynamicDialogRef | undefined;
  public pr: any;

  boolean = false;
  public pref13: boolean = false;
  public pref14: boolean = false;
  public tar: any;

  };
  public cheque = {
    Banco: '',
    Ncuenta: '',
    Ncheque: '',
    Nombre: '',
    Plaza: ''
  };

  public productos: Producto[];
  public productoElejido: Producto;
```

```
public clienteFrompuntoVen
```

```
boolean = false;
```

```
public previousUrl: string = "";
```

```
filteredTipo: any[] = [];
```

```
cdr: ChangeDetectorRef, private router: Router, private activatedRoute: ActivatedRoute, private
```

```
this.clienteFrompuntoVenta = this.activatedRoute.snapshot.queryParamMap.get('cliente');
```

```
this._cargardata.tarjcredito().pipe(take(1)).subscribe((resp: any) => {
```

```
this.tipo = r
```

```
this.filterByDay();
```

```
});
```

```
}
```

```
filterByDay() {
```

```
const dayOfWeek = new Date().getDay(); // Do
```

```
'd1', // Domingo
```

```
1: 'd2', // Lunes
```

```
2: 'd3', // Martes
```

```
3: 'd4', // Miércoles
```

```
};
```

```
const dayField = dayFieldMap[dayOfWeek];
```

```
this.filteredTipo = this.tipo.filter(item => {
```

```
this.ref.close();
```

```
}
```

```
}
```

```
ngOnInit() {
```

```
}
```

```
selectTipo(item: any) {
```

```
this.tipoVal = item.tarjeta; // Almacena el centro seleccionado
```

```
this.codTarj = item.cod_tarj
```

```
item.activadatos;
```

```
this.listaPrecioF(); // aca se llama a la funcion que muestra los prefijos
```

```
// aca se llama a la funcion que muestra los prefijos
```

```
}
```

```
else if (this.activaDatos
```

```

funcion que muestra los prefijos
    }
    else {
        this._cargardata.artsucursal().pipe(take(
        this.productos = [...resp.mensaje];
        // Forzar la detección de cambios
        this.cdr.d

        title: 'Ingrese los datos de la tarjeta',
        html: `<input type="text" id="titular" cla
id="dni" class="swal2-input" placeholder="DNI">
        <input type="number" id="numero" cla
type="number" id="autorizacion" class="swal2-input" placeholder="Autorización">`,
        showCan

cancelButtonText: 'Cancelar',
        preConfirm: () => {
            const titular = (<HTMLInputElement>document.getElementById('dni')).value;
            const numero = (<HTMLInputElement>document.getElementById('numero')).value;
            const autorizacion = (<HTMLInputElement>document.getElementById('autorizacion')).value;
            if (!
            Swal.showValidationMessage(`Por favor rellene todos los campos`);
            //return;
        }

        RegExp("[0-9]{8}$");
        let reTitular = new RegExp("[a-zA-Z ]{1,40}$");
        let reAut

        reNumero.test(numero)) {
            Swal.showValidationMessage(`El número de la tarjeta no es vá

            Swal.showValidationMessage(`El DNI no es válido`);
            //return;
        }

        Swal.showValidationMessage(`El titular no es válido`);
        //return;
    }
    if (
    Swal.showValidationMessage(`La autorización no es válida`);
    //return;
    }

```

```

    }).then((result) => {
        if (result.value) {
            this.tarjeta.Titular = result.value.
        }
    });

    this.tarjeta.Numero = result.value.numero;
    this.tarjeta.Autorizacion = result.value.autorizacion;

    this._cargardata.artsucursal().pipe(take(1)).subscribe((resp: any) => {
        console.log('Cargando sucursal');
    });

    // Forzar la detección de cambios
    this.cdr.detectChanges();
});

'Ingrese los datos del Cheque',
html: `
<input type="text" id="banco" class="swal2-input"
class="swal2-input" placeholder="N° Cuenta">
<input type="number" id="ncheque" class="swal2-input"
id="nombre" class="swal2-input" placeholder="Nombre">
<input type="text" id="plaza" class="swal2-input"
id="importeimputar" class="swal2-input" placeholder="Importe a Imputar">
<input type="number" id="nimporte" class="swal2-input"
Cheque">
<input type="text" id="fechacheque" class="swal2-input" placeholder="Fecha del
cheque" />

'date' para activar el datepicker nativo
document.getElementById('fechacheque').setAttribute('type', 'date');

confirmButtonText: 'Aceptar',
cancelButtonText: 'Cancelar',
preConfirm: () => {
    const banco = (<HTMLInputElement>document.getElementById('banco')).value;
    const ncuenta = (<HTMLInputElement>document.getElementById('ncheque')).value;
    const nombre = (<HTMLInputElement>document.getElementById('nombre')).value;
    const importeimputar = (<HTMLInputElement>document.getElementById('importeimputar')).value;
    const importecheque = (<HTMLInputElement>document.getElementById('importecheque')).value;
    const fechacheque = (<HTMLInputElement>document.getElementById('fechacheque')).value;
    if (!banco || !ncuenta || !nombre || !importeimputar || !importecheque || !fechacheque) {
        Swal.fire('Error', 'Por favor, complete todos los campos correctamente.', 'error');
    }
}

```

```

        Swal.showValidationMessage(`Por favor rellene todos los campos`);

{1,40}$");
        let reNcuenta = new RegExp("[0-9]{1,40}$");
        let reNcheque = new RegExp(
{1,40}$");
        let rePlaza = new RegExp("[a-zA-Z ]{1,40}$");
        let reImporteImputar =
RegExp("[0-9]{1,40}$");
        let reFechaCheque = new RegExp("\\d{2}/\\d{2}/\\d{4}$");//("
        if (!reBanco.test(banco)) {
            Swal.showValidationMessage(`El nombre del banco no
reNcuenta.test(ncuenta)) {
            Swal.showValidationMessage(`El numero de cuenta no es válido
        Swal.showValidationMessage(`El numero de cheque no es válido`);
        //return;

        Swal.showValidationMessage(`El nombre no es válido`);
        //return;
    }
    if (!
es válida`);
        //return;
    }
    if (!reImporteImputar.test(importeimputar)) {
válido`);
        //return;
    }
    if (!reImporteCheque.test(importecheque)) {
        //return;
    }
    return { banco, ncuenta, ncheque, nombre, plaza, importeim

    if (result.value) {
        this.cheque.Banco = result.value.banco;
        this.cheque.No
result.value.ncheque;
        this.cheque.Nombre = result.value.nombre;
        this.cheque.Plaz
result.value.importeimputar;
        this.cheque.ImporteCheque = result.value.importecheque;

```

```

console.log('Cheque guardado:', this.cheque);//console.log('Tarjeta guardada:', this.tarjeta);
any) => {
    console.log(resp.mensaje);
    this.productos = [...resp.mensaje];

listaPrecioF() {
    if (this.listaPrecio == "0") {
        this.prefi0 = true;
        this.prefi1 = false;
        this.prefi4 = false;
    }
    else if (this.listaPrecio == "1") {
        this.prefi0 = false;
        this.prefi1 = true;
        this.prefi4 = false;
    }
    else if (this.listaPrecio == "2") {
        this.prefi0 = true;
        this.prefi1 = false;
        this.prefi3 = false;
        this.prefi4 = false;
    }
    else if (this.listaPrecio == "3") {
        this.prefi0 = false;
        this.prefi1 = true;
        this.prefi3 = true;
        this.prefi4 = false;
    }
    else if (this.listaPrecio == "4") {
        this.prefi0 = true;
        this.prefi1 = true;
        this.prefi3 = false;
        this.prefi4 = true;
    }
}

selectProducto(
    producto,
    cliente: this.clienteFrompuntoVenta,
    tarjeta: this.tarjeta,
    cheque: this.cheque
);

```

```

        listaPrecio: this.listaPrecio,
    };
    this.ref = this.dialogService.open(Calculoproduccion);

    {
        producto: producto,
        cliente: this.clienteFrompuntoVenta,
        tarjeta: this.tarjeta,

        codTarj: this.codTarj,
        listaPrecio: this.listaPrecio,
    },
    contentStyle: 'border: 1px solid black; padding: 5px; width: 100%; height: 100%;'
));
}
exportExcel() {
    import('xlsx').then((xlsx) => {
        const worksheet = xlsx.utils.json_to_sheet(
            { data: worksheet }, SheetNames: ['data'] );
        const excelBuffer: any = xlsx.write(workbook, { bookType: 'xlsx', worksheet: 'data' });
        this.saveAsExcelFile(excelBuffer, 'products');
    });
}
saveAsExcelFile(buffer: any, fileName: string) {
    const blob = new Blob([buffer], { type: 'application/vnd.ms-excel' });
    const url = window.URL.createObjectURL(blob);
    const a = document.createElement('a');
    a.href = url;
    a.download = fileName + ".xlsx";
    a.click();
    window.URL.revokeObjectURL(url);
}
FileSaver.saveAs(data, fileName + '_export_' + new Date().getTime() + EXCEL_EXTENSION);
}
}

```

```
import { Component } from '@angular/core';
```

C
@Component

```
condicionventacab.component.html',
```

```
styleUrls: ['./condicionventacab.component.css']
```

})

export c

```
}
```


components\cuentacorriente\cuentacorriente.component.ts

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { Table } from 'primeng/table';

import { Cliente } from '.../interfaces/cliente';
import { ActivatedRoute } from '@angular/router';

import { CargardataService } from '.../services/cargardata.service';
import { ExcelService } from '.../services/excel.service';
import { first, take } from 'rxjs/operators';

@Component({
  selector: 'app-cuentacorriente',
  templateUrl: './cuentacorriente.component.html',
  styleUrls: ['./cuentacorriente.component.css']
})
export class CuentacorrienteComponent implements OnInit {
  clientes: Cliente[];
  public clienteElejido: Cliente;

  constructor(
    private cargardataService: CargardataService,
    private excelService: ExcelService,
    private route: ActivatedRoute
  ) {}

  ngOnInit(): void {
    this.cargardataService.getClientes().pipe(
      first(), take(1)
    ).subscribe(
      (resp) => {
        console.log(resp);
        this.clientes = resp.mensaje;
      }, (err) => { console.log(err); }
    );
  }

  exportarExcel(): void {
    this._router.navigate(['components/cabeceras'], { queryParams: { cliente: JSON.stringify(clienteElejido) } });
  }

  private _router: Router;

  const worksheet = excelService.json_to_sheet(this.clientes);
  const workbook = { Sheets: { 'Hoja1': worksheet } };
  const excelBuff = excelService.write(workbook, { bookType: 'xlsx', type: 'array' });
  this.saveAsExcelFile(excelBuff, 'cuentacorriente.xlsx');

  string): void {
    let EXCEL_TYPE = 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet';

    data: Blob = new Blob([buffer], {
      type: EXCEL_TYPE
    });

    FileSaver.saveAs(data, fileName);
  }
}
```

components\editcliente\editcliente.component.ts

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import {
  ReactiveFormsModule, AbstractControl, ValidatorFn, AbstractControlOptions, FormBuilder, FormGroup,
  Validators
} from '@angular/forms';
import { SubirdataService } from 'src/app/services/subirdata.service';
import { debounceTime } from 'rxjs';

@Component({
  selector: 'app-editcliente',
  templateUrl: './editcliente.component.html',
  styleUrls: ['./editcliente.component.css']
})
export class EditclienteComponent implements OnInit {
  public editarclienteForm!: FormGroup;
  public clienteForm!: FormGroup;
  public sucursal: string;
  public cuitFlag: boolean;
  public dniFlag: boolean;
  public telefonoFlag: boolean;

  constructor(private subirdata: SubirdataService, private router: Router, private activatedRoute: ActivatedRoute) {
    this.sucursal = localStorage.getItem('sucursal');
    this.clienteForm = this.activatedRoute.snapshot.data['clienteForm'];
    this.clienteForm = JSON.parse(this.clienteForm);
    console.log(this.clienteForm);

    this.monitorFormChanges();
  }

  ngOnInit(): void {
    this.inicializarForm();
    this.cargarDatosForm();
  }

  cuit: new FormControl(''),
  dni: new FormControl(''),
  telefono: new FormControl(''),
  ingresos_br: new FormControl(''),
  ingresos_neto: new FormControl('');

  inicializarForm() {
    this.editarclienteForm = this.formBuilder.group({
      nombre: this.clienteForm.nombre.trim(), Validators.compose([Validators.required, Validators.minLength(3)]),
      apellido: this.clienteForm.apellido.trim(), Validators.compose([Validators.required, Validators.minLength(3)]),
      sucursal: this.sucursal,
      cuit: this.clienteForm.cuit,
      dni: this.clienteForm.dni,
      telefono: this.clienteForm.telefono,
      ingresos_br: this.clienteForm.ingresos_br,
      ingresos_neto: this.clienteForm.ingresos_neto
    });
  }

  cargarDatosForm() {
    this.subirdata.getSubirdata().subscribe({
      next: (data) => {
        this.sucursal = data.sucursal;
      }
    });
  }
}
```

```

cuit: new FormControl(this.clienteFrompuntoVenta.cuit, Validators.compose([Validators.required

FormControl(this.clienteFrompuntoVenta.dni, Validators.compose([Validators.required,
Validators.p

FormControl(this.clienteFrompuntoVenta.telefono || 0, Validators.compose([
Validators.p

FormControl(this.clienteFrompuntoVenta.direccion.trim(), Validators.compose([Validators.required

])),
    tipoiva: new FormControl(this.clienteFrompuntoVenta.tipoiva),
    ingresos_br

onSelectionChange(event: any) {
    const selectedValue = event.target.value;
    console.log(se

this.editarclienteForm.controls['cuit'].setValue(0);
    }
    else {
        this.editarclienteForm

selectedValue
    }
    guardar(form: FormGroup) {
        if (form.valid) // si el formulario es valido

console.log("TIPO IVA:" + form.value.tipoiva);
        if (form.value.tipoiva == "Excento") {

"Monotributo") {
            cod_iva = 3;
        }
        else if (form.value.tipoiva == "Responsable I

(form.value.tipoiva == "Consumidor Final") {
            cod_iva = 2;
        }
        let date = new Da

+ "-" + date.getDate();//let fecha= date.getDate() + "/" + (date.getMonth()+ 1) + "/" +date.get

":" + date.getSeconds();
        let editadoCliente =
        {
            "cliente": parseInt(this.clienteFrompuntoVenta.cuit)

```

```
"direccion": form.value.direccion,
"dni": parseInt(form.value.dni),
"cuit": form.value.cuit,
parseInt(this.clienteFrompuntoVenta.cod_iva),
"cod_ven": parseInt(this.clienteFrompuntoVenta.cod_venta),
parseInt(this.clienteFrompuntoVenta.cod_zona),
"tipoiva": form.value.tipoiva,//this.clienteFrompuntoVenta.tipoiva,
this.clienteFrompuntoVenta.vendedor,
"zona": this.clienteFrompuntoVenta.zona,//"",

"idcli": parseInt(this.clienteFrompuntoVenta.idcli),
"id_cli": parseInt(this.clienteFrompuntoVenta.id_cliente),

"ingresos_br": form.value.ingresos_br,
"n_sucursal": this.clienteFrompuntoVenta.n_sucursal,
parseInt(this.clienteFrompuntoVenta.id_suc),
"estado_act": this.clienteFrompuntoVenta.estado_actual,

== "eliminar") {
    this.subirdata.eliminarCliente(editadoCliente, this.sucursal).subscribe(
        () => {
            Swal.fire({
                title: 'Eliminando...',
                timer: 300,
                didOpen: () => {
                    if (result.dismiss === Swal.DismissReason.timer) {
                        console.log('I was dismissed by the timer')
                    }
                }
            })
        },
        () => {
            else if (this accion == "guardar") {
                if (editadoCliente.cuit == "") {
                    Swal.fire({
                        title: 'ERROR',
                        text: 'Se requiere un cuit para este cliente'
                    })
                }
            }
        }
    )
}

true,
confirmButtonColor: '#3085d6',
cancelButtonColor: '#d33',
```

```

        this.subirdata.editarDatosClientes(editadoCliente, this.sucursal).subscribe((data: any) => {

            title: 'Guardando...',
            timer: 300,
            didOpen: () => {

            }).then((result) => {
                if (result.dismiss === Swal.DismissReason.timer) {

                    window.history.back();
                }
            })
        });
    }

    Swal.fire({
        title: 'ERROR',
        text: 'Verifique los datos ingresados, hay campos incorrectos',
        showCancelButton: true,
        confirmButtonColor: '#3085d6',
        cancelButtonColor: '#d354',
        confirmButtonText: 'Aceptar',
        cancelButtonText: 'Cancelar'
    });

    console.log("invalido");
    // Marca los campos como tocados para que se muestren los errores
    form.get(control).markAsTouched();
    }
    }
    monitorFormChanges() {
        Object.keys(this.editarclienteForm.controls).forEach((key) => {
            this.editarclienteForm.get(key).valueChanges.pipe(debounceTime(1000)).subscribe((value) => {
                this.nombreFlag = this.editarclienteForm.controls['nombre'].invalid;
                this.dniFlag = this.editarclienteForm.controls['dni'].invalid;
                this.telefonoFlag = this.editarclienteForm.controls['telefono'].invalid;
                this.direccionFlag = this.editarclienteForm.controls['direccion'].invalid;
            });
        });
    }
}

```

```

import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { ActivatedRoute, Router } from '@angular/router';
import { LoginService } from '.../services/login.service';
import { CryptoService } from '.../services/crypto.service';
import { CrudService } from '.../services/crud.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  form: FormGroup;
  public validacion: boolean = false;
  public mail: string;
  public check: boolean;
  errorhttp: string;
  public spinnerbotoningresar: boolean = true;
  public sucursal: string;
  _crypto: CryptoService, private _crud: CrudService) {
    // ...
  }

  ngOnInit() {
    if (localStorage.getItem('sddccasdf') != null) {
      let check: string = localStorage.getItem('sddccasdf');
      //login con datos guardados-----
    } else {
      this.check = false;
    }

    if (localStorage.getItem("sddeasdf") != null) {
      let email: any = localStorage.getItem('sddeasdf');
      email = this._crypto.decrypt(email);
      // ...
    }

    this.form = new FormGroup({
      email: new FormControl(null, [Validators.required, Validators.email]), // primer valor es
    });
  }
}

```

```

Validators.required)
    });
    }
    ingresar(f: any) {
        if (this.sucursal == null) {
            Swal.fire({
                title: 'Error!',
                text: 'No se puede ingresar sin sucursal',
                icon: 'error',
                showCancelButton: true,
                confirmButtonText: 'OK',
                cancelButtonText: 'Cancelar',
            });
        }
        this.sucursal = f.sucursal;
        this.checkPromise().then(() => {
            console.log("email:" + f.value.email);
            online-----
            this._login.SignIn(f.value.email, f.value.password);
            (resp == true) {
                this._crud.getListSnapFilter('usuarios/cliente', 'email', f.value.email);
                localStorage.setItem('usernameOp', resp2[0].payload.val().username);
                localStorage.setItem('sddffasdf', this._crypto.encrypt(resp2[0].payload.val().nivel)); // sddffasdf:pass, cdckcdck:true
                resp2[0].key; //sddggasdf userkey
                f.reset();
                this.router.navigate(['dashboard']);
            });
            if (this.check == true) {
                localStorage.setItem('sddccasdf', this._crypto.encrypt(f.value.password)); // sddccasdf:pass, cdckcdck:true
                localStorage.setItem('sddccasdf:check', cdckcdck:true);
                localStorage.setItem('sddccasdf', this._crypto.encrypt(f.value.password)); // sddccasdf:pass, cdckcdck:true
                localStorage.setItem('sddeasdf', this._crypto.encrypt(f.value.password)); // sddeasdf:pass, cdckcdck:true
                localStorage.setItem('sddccasdf', this._crypto.encrypt(f.value.password)); // sddccasdf:pass, cdckcdck:true
                localStorage.setItem('sddeasdf', this._crypto.encrypt(f.value.password)); // sddeasdf:pass, cdckcdck:true
                f.reset();
            }
            Swal.fire({
                title: 'Error!',
                text: 'Usuario incorrecto',
                icon: 'error',
                showCancelButton: true,
            });
        });
    }
}

```

```
'#d33',
```

```
confirmButtonText: 'OK',
```

```
});
```

```
}
```

```
}}
```

```
});
```

```
title: 'Input email address',
```

```
input: 'email',
```

```
inputLabel: 'Your email address'
```

```
{
```

```
  this._login.ForgotPassword(email);
```

```
}
```

```
}
```

```
checkPromise() {
```

```
  let promise = new Pr
```

```
localStorage.setItem('sddccasdf', "true");
```

```
}
```

```
else {
```

```
  localStorage.setItem('sdd
```

```
onSelectionChange(event: any) {
```

```
  this.sucursal = event.target.value;
```

```
}
```

```
}
```



```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import {
  ReactiveFormsModule, AbstractControl, ValidatorFn, AbstractControlOptions, FormBuilder, FormGroup,
  Validators
} from '@angular/forms';
import { SubirdataService } from 'src/app/services/subirdata.service';
import { CargardataService } from 'src/app/services/cargardata.service';
import { debounceTime, take } from 'rxjs/operators';

@Component({
  templateUrl: './newcliente.component.html',
  styleUrls: ['./newcliente.component.css']
})
export class NewclienteComponent implements OnInit {
  public clienteFrompuntoVenta: any;
  public sucursal: any;
  public nombreFlag: boolean;
  public direccionFlag: boolean;
  public cod_ivaFlag: boolean;
  public vendedores: any;
  public nombreVendedor: any;

  constructor(private _cargardata: CargardataService, private fb: FormBuilder) {
    this.cargarForm();
    this.monitorearForm();
  }

  this.nuevoclienteForm = this.fb.group({
    nombre: new FormControl('', Validators.compose([Validators.required, Validators.pattern(/^[a-zA-Z0-9]{1,15}$/)])),
    cuit: new FormControl('', Validators.compose([Validators.required, Validators.pattern(/^[0-9]{5,15}$/)])),
    direccion: new FormControl('', Validators.compose([Validators.required, Validators.pattern(/^[a-zA-Z0-9]{2,60}$/)]))
  });

```

```

    ])),
    cod_iva: new FormControl('', Validator

    },,);
    }
    onSelectionVendedorChange(event: any) {
        this.codigoVendedor = this.vendedores[
this.vendedores[event.target.value].vendedor;
    }

    onSelectionChange(event: any) {
        const

if (selectedValue == "2") {
    this.nuevoclienteForm.controls['cuit'].setValue(0);
}

// Realiza acciones basadas en selectedValue
    }
    guardar(form: FormGroup) {
        if (form.va

= ["", "Responsable Inscripto", "Consumidor Final", "Monotributo", "Excento"];
        let indexn

genera un numero entre 10000 y 9999999 incluidos
        let sucursal: any = localStorage.getItem

Math.floor((Math.random() * 99999) + 10000);
        let date = new Date();
        let fecha = date

let fecha= date.getDate() + "/" + (date.getMonth()+ 1) + "/" +date.getFullYear();
        let hora

let nuevoCliente =
    {
        "cliente": (sucursal * 100000) + nuevoclirandom,//parse

"direccion": form.value.direccion,
        "dni": parseInt(form.value.dni),
        "cuit": form

"cod_ven": this.codigoVendedor,//this.vendedor.cod_ven,//parseInt(form.value.cod_ven),

ivaArray[form.value.cod_iva],// ""
        "vendedor": this.nombreVendedor,//this.vendedor.vend

form.value.telefono,
        "estado": "",//"",
        "idcli": indexnuevocli,//parseInt(form.v

parseInt(form.value.id_cli),
        "fecha": fecha,
        "hora": hora,
        "ingresos_br":

```

```

        "id_suc": indexnuevocli,
        "estado_act": ""
    }
    if (nuevoCliente.cuit == 0 && nue

title: 'ERROR',
    text: 'Se requiere un cuit para este tipo de IVA',
    icon: 'er

'#3085d6',
    cancelButtonColor: '#d33',
    cancelButtonText: 'Cancelar'
    })

this.subirdata.subirDatosClientes(nuevoCliente, sucursal).subscribe((data: any) => {

'Guardando...',
    timer: 300,
    didOpen: () => {
        Swal.showLoadi

(result.dismiss === Swal.DismissReason.timer) {
    console.log('I was closed by the

    });
    }
    }
    else {
        this.monitorFormChanges();
        //mostrar en consola qu

console.log(form);
    Swal.fire({
        title: 'ERROR',
        text: 'Verifique los datos i

        icon: 'error',
        showCancelButton: true,
        confirmButtonColor: '#3085d6',

        console.log("invalido");
        // Marca los campos como tocados para que se muestren los

form.get(control).markAsTouched();
    }
}

```

```

        }
        cargarVendedor() {
            this._cargardata.vend
        }

        console.log(resp);
        this.vendedores = resp.mensaje;
    });
}

monitorFormChanges() {

    const control = this.nuevoclienteForm.get(field);
    control.valueChanges.pipe(debounceTime(500))
        .subscribe(() => {
            cambió a: `, value);
            this.nombreFlag = this.nuevoclienteForm.controls['nombre'].invalid;

            this.dniFlag = this.nuevoclienteForm.controls['dni'].invalid;
            this.telefonoFlag = this.nuevoclienteForm.controls['telefono'].invalid;

            this.direccionFlag = this.nuevoclienteForm.controls['direccion'].invalid;
        });
    });
}
}

```

```

import { Component, OnInit } from '@angular/core';
//import { CargardataService } from '../serv

declare function init_plugins();

@Component({
  selector: 'app-p

pages.component.css']
})
export class PagesComponent implements OnInit {

this._cargardata.tarjcredito().pipe(take(1)).subscribe((resp:any)=>{
  console.log(resp.men

JSON.stringify(resp.mensaje));
  localStorage.setItem('lastSelectedValue', JSON.stringify(

console.log("PAGES COMPONENTS");
  init_plugins();
}
}

```

components\pedidos\pedidos.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';

import './pedidos.component';

describe('PedidosComponent', () => {
  let component: PedidosComponent;
  let fixture: ComponentFixture<PedidosComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ PedidosComponent ]
    }).compileComponents();

    fixture = TestBed.createComponent(PedidosComponent);
    component = fixture.componentInstance;

    expect(component).toBeTruthy();
  });
});
```

components\pedidos\pedidos.component.ts

```
import { Component, OnInit } from '@angular/core';
import { DynamicDialogConfig, DynamicDialogRef } from '@ng-dynamic-dialog/core';

import { CargardataService } from 'src/app/services/cargardata.service';

@Component({
  selector: 'app-pedidos',
  templateUrl: './pedidos.component.html',
  styleUrls: ['./pedidos.component.css']
})
export class PedidosComponent implements OnInit {
  public ref: DynamicDialogRef;
  public config: DynamicDialogConfig;

  cargardataService: CargardataService;

  ngOnInit() {
    this.cargardataService = this.config.data;
    this.cargardataService.pedidoxComprobante(cod_sucursal, cod_documento);

    this.pedidos = data.mensaje;
  }
}
```

components\productos\productos.component.ts

```
import { Component } from '@angular/core';
import Swal from 'sweetalert2';

@Component({
  selector: 'app-productos',
  templateUrl: './productos.component.html',
  styleUrls: ['./productos.component.css']
})
export class ProductosComponent {
  public estado: string;

  // Función para abrir el modal
  async openModal() {
    const { value } = await Swal.fire({
      title: '¿Estás seguro?',
      text: '¿Quieres abrir el modal?',
      icon: 'warning',
      showCancelButton: true,
      confirmButtonText: 'OK',
      cancelButtonText: 'Cancelar'
    });

    if (value) {
      this.abrirModal();
    }
  }

  // Supongamos que tienes los siguientes estados
  estados = {
    'estado1': 'Estado 1',
    'estado2': 'Estado 2',
    'estado3': 'Estado 3'
  };
}
```



```
import { Component, OnInit, ViewChild } from '@angular/core';
import { Table } from 'primeng/table';
import { Cliente } from '.../interfaces/cliente';
import { ActivatedRoute } from '...';
import { first, take } from 'rxjs/operators';

templateUrl: './puntoventa.component.html',
styleUrls: ['./puntoventa.component.css']
export class PuntoVentaComponent implements OnInit {
  public clientes: Cliente[];
  public clienteElejido: Cliente;
  constructor(private _cargardata: CargardataService) {}

  let sucursal: string = localStorage.getItem('sucursal');
  this._cargardata.clisucx(sucursal).subscribe(
    (resp) => {
      this.clientes = resp.mensaje;
    }, (err) => { console.log(err); }
  );
  selectCliente(cliente: Cliente) {
    this._router.navigate(['components/condicionventa'], { queryParams: { cliente: JSON.stringify(cliente) } });
  }
  this._router.navigate(['components/editcliente'], { queryParams: { cliente: JSON.stringify(cliente) } });
}

const worksheet = xlsx.utils.json_to_sheet(this.clientes);
const workbook = { Sheets: { 'condicionventa': worksheet } };
xlsx.write(workbook, { bookType: 'xlsx', type: 'array' });
this.saveAsExcelFile(excelBuffer, 'condicionventa.xlsx');

string): void {
  let EXCEL_TYPE = 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet';
  data: Blob = new Blob([buffer], { type: EXCEL_TYPE });
  FileSaver.saveAs(data, fileName);
}
```


components\recibos\recibos.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';

import {
  RecibosComponent
} from './recibos.component';

describe('RecibosComponent', () => {
  let component: RecibosComponent;
  let fixture: ComponentFixture<RecibosComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ RecibosComponent ]
    }).compileComponents();

    fixture = TestBed.createComponent(RecibosComponent);
    component = fixture.componentInstance;

    expect(component).toBeTruthy();
  });
});
```

components\recibos\recibos.component.ts

```
import { Component, OnInit } from '@angular/core';
import { DynamicDialogConfig, DynamicDialogRef } from '@angular/material/dialog';

import { CargardataService } from 'src/app/services/cargardata.service';

@Component({
  selector: 'app-recibos',
  templateUrl: './recibos.component.html'
})
export class RecibosComponent implements OnInit {
  recibos: any[] = [];
  errorMessage: string;

  constructor(
    private dialogRef: DynamicDialogRef,
    private config: DynamicDialogConfig,
    private cargardataService: CargardataService
  ) {}

  ngOnInit(): void {
    this.config.data;
    this.cargarDataService.reciboxComprobante(cod_sucursal, comprobante).subscribe(
      (data) => {
        this.errorMessage = data.mensaje;
        console.error(this.errorMessage);
      },
      () => {
        this.errorMessage = 'Ocurrió un error al cargar los datos.';
        console.error(this.errorMessage);
      }
    );
  }
}
```

components\venta\venta.component.ts

```
import { Component } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';

selector: 'app-venta',
templateUrl: './venta.component.html',
styleUrls: ['./venta.component.css'],

carritoArray: any[] = [];

constructor(private activatedRoute: ActivatedRoute) {
  this.carritoString = this.activatedRoute.snapshot.paramMap.get('carritoString');
  this.carritoArray = JSON.parse(this.carritoString);
  console.log(this.carritoArray);
}
```

config

[illegible]

//motomatch loc

```
index.php/Login";Ð
    Ð
    Ð
    //CARGA-----

index.php/Carga/Arconmov";//clientes clisucÐ
    export const Urlartsucursal= "https://motomatch.segu239.com/APIAND/index.php/Carga/Arconmov";Ð
export const Urltarjcredito="https://motomatch.segu239.com/APIAND/index.php/Carga/Tarjcredito";Ð
index.php/Carga/Vendedores";Ð
    export const Urlclisucx="https://motomatch.segu239.com/APIAND/index.php/Carga/Vendedores";Ð
Urlpedidox="https://motomatch.segu239.com/APIAND/index.php/Carga/Pedidox";Ð
    export const Urlcabecera="https://motomatch.segu239.com/APIAND/index.php/Carga/Pedidox";Ð

//DESCARGA-----Ð
    export const Urlclisucxapp="https://motomatch.segu239.com/APIAND/index.php/Carga/Pedidox";Ð
Clisucxapp";//Ð
    export const UpdateClisucxappWeb= "https://motomatch.segu239.com/APIAND/index.php/Carga/Pedidox";Ð
motomatch.segu239.com/APIAND/index.php/Descarga/Pedidossucxapp";Ð
    export const Urlpedidossucxapp="https://motomatch.segu239.com/APIAND/index.php/Descarga/Pedidossucxapp";Ð
PedidossucxappCompleto";Ð
    export const Urlarticulossucxapp="https://motomatch.segu239.com/APIAND/index.php/Descarga/Pedidossucxapp";Ð
motomatch.segu239.com/APIAND/index.php/Descarga/Mixto";Ð
    export const UrlclisucxappWeb= "https://motomatch.segu239.com/APIAND/index.php/Descarga/Mixto";Ð
export const UrleliminarCliente= "https://motomatch.segu239.com/APIAND/index.php/Descarga/eliminarCliente";Ð
motomatch.segu239.com/APIAND/index.php/Descarga/UpdateArtsucxapp";//Ð
    export const UpdateArtsucxappManagedPHP="https://motomatch.segu239.com/APIAND/index.php/Descarga/UpdateArtsucxappManagedPHP";Ð
Descarga/UpdateArtsucxappManagedPHP";// *Ð
    Ð
    //----- tailscaleÐ
    export const UrlclisucxappWeb= "https://motomatch.segu239.com/APIAND/index.php/Descarga/UpdateArtsucxappManagedPHP";Ð

CARGA-----Ð
    export const Urlartsucursal= "http://100.65.39.89/APIAND/index.php/Carga/Artsucursal";Ð
export const Urltarjcredito="http://100.65.39.89/APIAND/index.php/Carga/Tarjcredito";Ð
    export const Urlvendedores="http://100.65.39.89/APIAND/index.php/Carga/Vendedores";Ð
Clisucx";Ð
    export const Urlpedidox="http://100.65.39.89/APIAND/index.php/Carga/Pedidox";Ð
    export const Urlcabecera="http://100.65.39.89/APIAND/index.php/Carga/Pedidox";Ð
PedidoxComprobante";Ð
    export const UrlreciboxComprobante="http://100.65.39.89/APIAND/index.php/Carga/Pedidox";Ð
APIAND/index.php/Carga/Cabecerax";Ð
    export const Urlcabecerasuc="http://100.65.39.89/APIAND/index.php/Carga/Pedidox";Ð
UrlcabeceraLastId="http://100.65.39.89/APIAND/index.php/Carga/LastIdnum";Ð
    Ð
    //DESCARGA-----
```

```
Urlclisucxapp= "http://100.65.39.89/APIAND/index.php/Descarga/Clisucxapp";Đ
                                                                    export const UpdateC
UpdateClisucxapp";Đ
    export const Urlpedidossucxapp="http://100.65.39.89/APIAND/index.php/Descarg
UrlpedidossucxappCompleto="http://100.65.39.89/APIAND/index.php/Descarga/PedidossucxappCompleto
index.php/Descarga/Articulossucxapp";Đ
    export const Urlmixto="http://100.65.39.89/APIAND/index.p
APIAND/index.php/Descarga/ClisucxappWeb";Đ
    export const UrlPagoCabecera= "http://100.65.39.89/AP
"http://100.65.39.89/APIAND/index.php/Descarga/eliminarCliente";Đ
    Đ
    export const UpdateArtsucxappW
export const UpdateArtsucxappWebManagedPHP= "http://100.65.39.89/APIAND/index.php/Descarga/Upd
```



```

import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, Router } from '@angular/router';
import { Observable } from 'rxjs';
import { LoginService } from '../services/login.service';
import { CryptoService } from '../services/crypto.service';

@Injectable({
  providedIn: 'root'
})
export class AdminGuard implements CanActivate {

  private a:boolean=false;

  constructor(private router:Router, private loginService:LoginService, private cryptoService:CryptoService) {}

  canActivate(): boolean {
    let a=localStorage.getItem('sddffasdf');
    if(a) {
      a= this._crypto.decrypt(a);
    }
    else {
      {
        this.login=false;
        this.router.navigate(['../components/noway']);
      }
    }
    return a;
  }
}

```

```
import { Injectable } from '@angular/core';
import { ActivatedRouteSnapshot, CanActivate, RouterStateSnapshot } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {
  canActivate(
    routerStateSnapshot: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {
```

```
import { Injectable } from '@angular/core';
import { ActivatedRouteSnapshot, CanActivate, RouterStateSnapshot } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class LogicaGuard implements CanActivate {
  canActivate(
    state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean {
```

```

import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, Router } from '@angular/router';
import { Observable } from 'rxjs';
import { LoginService } from '../services/login.service';
import { CryptoService } from '../services/crypto.service';

@Injectable({
  providedIn: 'root'
})
export class LoginguardGuard implements CanActivate {

  login:boolean=false;
  private a:boolean=false;

  constructor(private router:Router, private _loginService:LoginService, private _cryptoService:CryptoService) {}

  canActivate():boolean {

    let a=localStorage.getItem('sddffasdf');
    if(a) {
      a= this._crypto.decrypt(a);
      if(a) {
        return true;
      }
    }
    else {
      {
        this.login=false;
        this.router.navigate(['/login']);
      }
    }
    return this.login;
  }

  // this.promise().then((resp:any)=>{
  //   if (resp != "fallo")
  //   {
  //     localStorage.setItem('logica',logica);
  //   }
  //   return false;
  // },()=>{return false} );
  // return new Promise((resolve, reject) => {
  this._crud.getListSnap('configuraciones').pipe(take(1)).subscribe((resp:any)=>{
  //   let logica=resp.logica;
  localStorage.setItem('logica',logica);
  //   resolve(true);
  // }, (error)=>{reject(error)});

```

// }

}

guards\super.guard.ts

```
import { Injectable } from '@angular/core';
import { ActivatedRouteSnapshot, CanActivate, RouterStateSnapshot } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class SuperGuard implements CanActivate {
  canActivate(
    routerStateSnapshot: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {
```

```
export interface Cabecera {
    tipo?: string;
    numero_int?: number;
    puntoventa?: number;

    anumero_com?: number;
    cliente?: number;
    cod_sucursal?: string;
    emitido?: Date;
    venta?: number;

    iva2?: number;
    iva3?: number;
    bonifica?: number;
    interes?: number;
    saldo?: number;

    cod_vendedor?: number;
    anulado?: boolean;
    cuit?: number;
    usuario?: string;

    imp_int?: number;
    fec_proceso?: Date;
    fec_ultpago?: Date;
    estado?: string;
    id_aso?: number;
}
```

```
export interface CarritoItem {
  cliente?:number;
  nombre?:string;
  direccion?:string;

  cod_zona?:number;
  tipoiva?:string;
  vendedor?:string;
  zona?:string;
  telefono?:string;

  fecha?:any;
  hora?:any;
}
```


interfaces\cliente.ts

```
export interface Cliente {
  cliente?: number;
  nombre?: string;
  direccion?: string;
  dni?: string;
  cod_zona?: number;
  tipoiva?: string;
  vendedor?: string;
  zona?: string;
  telefono?: string;
  fecha?: any;
  hora?: any;
}
```

interfaces\producto.ts

```
export interface Producto {
    nomart?:string;
    marca?:string;
    precon?:number;
    prefil?:number;
    exi1?:number;
    exi2?:number;
    exi3?:number;
    exi4?:number;
    exi5?:number;
    idart?:number;
}
```

```
export interface Recibo {
  recibo: number;
  c_tipo: string;
  c_numero: number;
  c_cuota: number;
  observacion: number;
  cod_lugar: string;
  sesion: number;
  c_tipf: string;
  c_puntoven: number;
  cod_sucursal: number;
  fec_proceso?: any;
  bonifica: number;
  interes: number;
  id_factura: number;
}
```

```
import { Pipe, PipeTransform } from '@angular/core';
```

```
PipeTransform {
```

```
    transform(items: any[], searchText: string): any[] {  
        if (!items) { return []; }
```

```
        searchText.toLowerCase();
```

```
        return items.filter( it => {
```

```
            return it.tarjeta.toLowerCase().indexOf(searchText.toLowerCase()) > -1; }  
        }
```

```

import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { HttpClient } from '@angular/common/http';

import {
  UrlreciboxComprobante, UrlpedidoxComprobante, Urlarconmov, Urlartsucursal, Urltarjcredito, Urlcl
  UrlcabeceraLastId, UrlPagoCabecera } from '../config/ini'
import { map } from "rxjs/operators";

export class CargardataService {

  constructor(private http:HttpClient) {}

  return this.http.get(Urlarconmov); //HTTP CLIENT DEVUELVE UN JSON POR DEFECTO, NO HACE FALLA

}

artsucursal() {
  return this.http.get(Urlartsucursal);
}

tarjcredito() {
  return this.http.get(Urltarjcredito);
}

this.http.get(Urlvendedores);
}
clisucx(cod_sucursal:string) // asi es como funcionaba con un servicio
{
  "sucursal": cod_sucursal
  })
}

pedidox(cod_sucursal:string)
{
  return this.http.get(Urlpedidox);
}

pedidoxComprobante(cod_sucursal:string, comprobante:any)
{
  return this.http.post(UrlpedidoxComprobante, comprobante);
}

comprobante
{
  return this.http.post(UrlpedidoxComprobante, comprobante);
}

reciboxComprobante(cod_sucursal:string, comprobante:any)
{
  return this.http.post(UrlreciboxComprobante, comprobante);
}

```

```

        "comprobante": comprobante
    })
    }
    // cabeceras por sucursal
    cabecerasuc(cod_sucursal:s
{
    return this.http.post(Urlcabecerasuc,{
        "sucursal": cod_sucursal
    })
}
// cabecera
como funcionaba con una tabla de cliente por sucursal
{
    return this.http.post(Urlcabecerax
lastIdnum(cod_sucursal:string) // asi es como funcionaba con una tabla de cliente por sucursal
cod_sucursal
    })
}
    pagoCabecera(cod_sucursal:string, pagoCC:any) // asi es como fu
this.http.post(UrlPagoCabecera,{
    "sucursal": cod_sucursal,
    "pagoCC": pagoCC
})

```

```

import { Injectable } from '@angular/core';
import { BehaviorSubject } from 'rxjs';

class CarritoService {
  private itemsEnCarrito: any[] = [];
  private carritoSubject = new BehaviorSubject<any[]>([]);

  constructor() {
    this.actualizarCarrito();
  }

  get carritoObservable() {
    return this.carritoSubject;
  }

  const carritoData = localStorage.getItem('carrito');
  if (carritoData) {
    this.itemsEnCarrito = JSON.parse(carritoData);
  }

  this.carritoSubject.next(this.itemsEnCarrito);
}

// Agregar otros métodos para modificar el carrito

agregarItemCarritoJson(clave: string, valor: any): void {
  // Obtener el valor actual para la clave
  let array: any[] = [];
  if (items) {
    // Si la clave existe, convertir de JSON a array
    array = JSON.parse(items);
  }
  array.push(valor);
  // Guardar el array actualizado en localStorage como una cadena de JSON
  localStorage.setItem(clave, JSON.stringify(array));
  this.actualizarCarrito();
}

```

services\crud.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { AngularFireDatabase } from '@angular/fire/compat/database';
import { first, take } from 'rxjs/operators';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class CrudService {

  constructor(private http: HttpClient, private db: AngularFireDatabase) {

  }

  getListSnap(item: string) {
    this.db.list(item).snapshotChanges();
    // getListSnapFilter(item: string, field: any, filter: any) {
    ref => ref.orderByChild(field).equalTo(filter)).snapshotChanges();
    // getListSnapFilterStart(item: string, field: any, filter: any) {
    usado para devolver rubros
    {
      return this.db.list(item, ref => ref.orderByChild(field).startAt(0)).snapshotChanges();
    }
    getListStateLastOne(item: string) {
      return this.db.list(item, ref => ref.limitToLast(1)).snapshotChanges();
    }
    this.db.list(item).push(JSON.parse(data));
    }
    update(item: string, key: string, data: string) {
      return this.db.update(item, key, data);
    }
    remove(item: string, key: string) {
      return this.db.remove(item, key);
    }
    // getNumeroSecuencial() {
    this.db.object('indices/operacion/' + key).valueChanges().pipe(
      map((data: any) => data.secuencial));
    nuevoSecuencial: number): Promise<void> {
      return this.db.object('indices/operacion/' + key).update({
        secuencial: nuevoSecuencial
      });
    }
  }
}
```


services\crypto.service.ts

```
import { Injectable } from '@angular/core';
import * as CryptoJS from 'crypto-js';

class CryptoService {
    private secretKey = "hjddfwmiytesfggdc";

    constructor() { }

    encrypt(value: string): string {
        return CryptoJS.AES.encrypt(value, this.secretKey.trim()).toString();
    }

    decrypt(textToDecrypt: string): string {
        return CryptoJS.AES.decrypt(textToDecrypt, this.secretKey.trim()).toString(CryptoJS.enc.Utf8);
    }
}
```

services\login.service.ts

```
import { Injectable } from '@angular/core';
import { AngularFireAuth } from '@angular/fire/compat/auth';
import { AngularFireDatabase } from '@angular/fire/compat/database';
import { Observable, Subject } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class LoginService {
  // Save logged in user data
  userData: any;

  // Inject Firebase auth service
  afAuth: AngularFireAuth;

  // Inject Firebase database
  private db: AngularFireDatabase;

  constructor(afAuth: AngularFireAuth, db: AngularFireDatabase) {
    this.afAuth = afAuth;
    this.db = db;
  }

  login(email: string, password: string, nivel: string): Observable<any> {
    return this.db.list('usuarios/cliente').valueChanges().pipe(
      map((users: any[]) => {
        const user = users.find((u) => u.email === email);
        if (user) {
          if (user.password === password) {
            if (user.nivel === nivel) {
              return user;
            } else {
              return null;
            }
          } else {
            return null;
          }
        } else {
          return null;
        }
      })
    );
  }

  getAdminUserByItem(item: any, child: string, value: string) // usado para devolver email
  {
    ref.orderByChild(child).equalTo(value)); // /comerciales');
  }

  getClientUserByItem(item: any, child: string, value: string) // usado para devolver email
  {
    ref.orderByChild(child).equalTo(value)); // /comerciales');
  }

  return this.response = this.db.list(item, ref => ref.orderByChild(child).equalTo(value)); // usado para devolver email
  {
    ref.orderByChild(child).equalTo(value)); // /comerciales');
  }

  this.db.list(item).snapshotChanges();
}

//firebase-----
SignIn(email: string, password: string, nivel: string) {
  return this.afAuth.signInWithEmailAndPassword(email, password)
    .then((result: any) => {
      if (result) {
        this.userData = result.user;
        return result;
      } else {
        return null;
      }
    });
}
```

```

        return true;
    }
    else {
        return
    }

    error.message;//window.alert(error.message)
    })
    }

    // Sign up with email/password
    Sig

    this.afAuth.createUserWithEmailAndPassword(email, password)
        .then((result: any) => {

        up and returns promise */
        this.SendVerificationMail();

    this.SetUserData(result.user);
    }).catch((error: any) => {
        return error.message;//w

    when new user sign up
        SendVerificationMail() {
            return this.afAuth.currentUser//.sendEmailV

        //this.router.navigate(['verify-email-address']);
        })
    }

    // Reset Forggot pass

    this.afAuth.sendPasswordResetEmail(passwordResetEmail)
        .then(() => {
            window.alert(

        }).catch((error) => {
            window.alert(error)
        })
    }

    // Returns true when use

    const user = JSON.parse(localStorage.getItem('user'));
    return (user !== null && user.email

    }
    setUserData(apellido: string, email: string, nombre: string, password: string, telefono:

```

apellido: apellido,

email: email,

nivel: "none",

nombre: nombre,

password:

this.afAuth.signOut();

}
}

services\motomatch-bot.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

constructor(private http: HttpClient) { }

sendTo

formData: FormData = new FormData();
formData.append('chat_id', '-4123033554');
formData

api.telegram.org/bot6654673391:AAEgy7peYqnm-OPjzavYo_D7PFv2IIIfq8/sendDocument`, formData).su

}
}
```

services\subirdata.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/c

UpdateArtsucxappWeb, Urlclisucxapp, Urlpedidossucxapp, UrlpedidossucxappCompleto, Urlarticulos
UrleliminarCliente } from '../config/ini';

@Injectable()

constructor(private http: HttpClient) { }

editarStockArtSucx(idart: number, suc: number, op:

"idart": idart,
    "exi": suc,
    "op": op, //"+", "-"
    });
    }

editarStockArtSu

this.http.post(UpdateArtsucxappWebManagedPHP,
    {
        "result": result,
        "exi": su

any) {
    return this.http.post(UpdateClisucxappWeb,
        {
            "clientes": data,
            "i

return this.http.post(UrlclisucxappWeb,
    {
        "clientes": data,
        "id_vend": id

console.log(data);
    console.log(id);
    return this.http.post(UrlpedidossucxappCompleto,

    "id_vend": id

    });
    }

    subirDatosArticulos(data: any, id: any) {
        return
```

"id_vend": id

});
}

subirDatosMixto(data: any, id: any)

"id_vend": id

});
}

eliminarCliente(data: any, id: any)

"clientes": data,

"id_vend": id

});
}
}

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-footer',
  templateUrl: 'app-footer.html',
  styleUrls: ['app-footer.component.css']
})
export class FooterComponent implements OnInit {
  constructor() { }
  ngOnInit(): void {
    // ...
  }
}
```



```

import { Component, OnInit } from '@angular/core';
import { CarritoService } from 'src/app/serv

@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['.

{
  public sucursal: string = '';
  public cantidad: number = 0;
  public sucursalNombre: string = '';

constructor(private carritoService: CarritoService) {
  this.sucursal = localStorage.getItem('s

Central';
  }
  else if (this.sucursal == '2') {
    this.sucursalNombre = 'Suc. Valle Viejo';
  }

Guemes';
  }
  else if (this.sucursal == '4') {
    this.sucursalNombre = 'Deposito';
  }
}

ngOnInit() {

any[]) => {
  this.cantidad = items.length;
  console.log(items);
  });
}
}

```

```

import { Component, OnInit } from '@angular/core';
import {LoginService} from '../../services/loginService';

export class SidebarComponent implements OnInit {
  templateUrl: 'sidebar.component.html',
  styleUrls: ['./sidebar.component.css']
}

export class SidebarComponent implements OnInit {
  _login:LoginService
  {
    this.nombreCliente=localStorage.getItem('usernameOp');
  }
  ngOnInit() {
    localStorage.setItem('sddddasdf',null);
    localStorage.setItem('sddeasdf',null);
    localStorage.setItem('sddeasdf',null);
    this._login.signOut().then((resp:any)=>console.log(resp));
  }
}

```

Table of Contents

[app.component.html](#)

..... [object Object]

[components\ analisiscaja\ analisiscaja.component.html](#)

..... [object Object]

[components\ analisispedidos\ analisispedidos.component.html](#)

..... [object Object]

[components\ cabeceras\ cabeceras.component.html](#)

..... [object Object]

[components\ calculoproducto\ calculoproducto.component.html](#)

..... [object Object]

[components\ carrito\ carrito.component.html](#)

..... [object Object]

[components\ condicionventa\ condicionventa.component.html](#)

..... [object Object]

[components\ condicionventacab\ condicionventacab.component.html](#)

..... [object Object]

[components\ cuentacorriente\ cuentacorriente.component.html](#)

..... [object Object]

[components\ editcliente\ editcliente.component.html](#)

..... [object Object]

[components\ login\ login.component.html](#)

..... [object Object]

[components\ newcliente\ newcliente.component.html](#)

..... [object Object]

[components\ pages.component.html](#)

..... [object Object]

[components\ pedidos\ pedidos.component.html](#)

..... [object Object]

[components\ productos\ productos.component.html](#)

..... [object Object]

[components\ puntoventa\ puntoventa.component.html](#)

..... [object Object]

[components\ recibos\ recibos.component.html](#)

..... [object Object]

[components\ venta\ venta.component.html](#)

..... [object Object]

[shared\ footer\ footer.component.html](#)

..... [object Object]

[shared\ header\ header.component.html](#)

..... [object Object]

[shared\ sidebar\ sidebar.component.html](#)

..... [object Object]

[components\ analisiscaja\ analisiscaja.component.css](#)

..... [object Object]

[components\ cabeceras\ cabeceras.component.css](#)

..... [object Object]

[components\ carrito\ carrito.component.css](#)

..... [object Object]

[components\ condicionventa\ condicionventa.component.css](#)

..... [object Object]

[components\ login\ login.component.css](#)

..... [object Object]

[app-routing.module.ts](#)

..... [object Object]

[app.component.ts](#)

..... [object Object]

[app.module.ts](#)

..... [object Object]

[components\ analisiscaja\ analisiscaja.component.ts](#)

..... [object Object]

[components\ analisispedidos\ analisispedidos.component.ts](#)

..... [object Object]

[components\ cabeceras\ cabeceras.component.ts](#)

..... [object Object]

[components\ calculoproducto\ calculoproducto.component.ts](#)

..... [object Object]

[components\ carrito\ carrito.component.ts](#)

..... [object Object]

[components\ condicionventa\ condicionventa.component.ts](#)

..... [object Object]

[components\ condicionventacab\ condicionventacab.component.ts](#)

..... [object Object]

[components\ cuentacorriente\ cuentacorriente.component.ts](#)

..... [object Object]

[components\ editcliente\ editcliente.component.ts](#)

..... [object Object]

[components\ login\ login.component.ts](#)

..... [object Object]

[components\ newcliente\ newcliente.component.ts](#)

..... [object Object]

[components\ pages.component.ts](#)

..... [object Object]

[components\ pedidos\ pedidos.component.spec.ts](#)

..... [object Object]

[components\ pedidos\ pedidos.component.ts](#)

..... [object Object]

[components\productos\productos.component.ts](#)

..... [object Object]

[components\puntoventa\puntoventa.component.ts](#)

..... [object Object]

[components\recibos\recibos.component.spec.ts](#)

..... [object Object]

[components\recibos\recibos.component.ts](#)

..... [object Object]

[components\venta\venta.component.ts](#)

..... [object Object]

[config\ini.ts](#)

..... [object Object]

[guards\admin.guard.ts](#)

..... [object Object]

[guards\auth.guard.ts](#)

..... [object Object]

[guards\logica.guard.ts](#)

..... [object Object]

[guards\login.guard.ts](#)

..... [object Object]

[guards\super.guard.ts](#)

..... [object Object]

[interfaces\cabecera.ts](#)

..... [object Object]

[interfaces\carritoItem.ts](#)

..... [object Object]

[interfaces\cliente.ts](#)

..... [object Object]

[interfaces\producto.ts](#)

..... [object Object]

[interfaces\recibo.ts](#)

..... [object Object]

[pipes\filter.pipe.ts](#)

..... [object Object]

[services\cargardata.service.ts](#)

..... [object Object]

[services\carrito.service.ts](#)

..... [object Object]

[services\crud.service.ts](#)

..... [object Object]

[services\crypto.service.ts](#)

..... [object Object]

[services\login.service.ts](#)

..... [object Object]

[services\motomatch-bot.service.ts](#)

..... [object Object]

[services\subirdata.service.ts](#)

..... [object Object]

[shared\footer\footer.component.ts](#)

..... [object Object]

[shared\header\header.component.ts](#)

..... [object Object]

[shared\sidebar\sidebar.component.ts](#)

..... [object Object]