Universidad ORT Uruguay

Facultad de Ingeniería

PROGRAMACIÓN 1

Obligatorio 1

Integrantes:

Joaquín Carrasco - Nro. Estudiante – 163609

Gastón Barlocco - Nro. Estudiante – 241025

Docentes: Sebastián Pesce, Guillermo Vieira

2020

Index.html

```
<!--
    Autores: Joaquín Carrasco (163609) & Gastón Barlocco (241025)
 -->

<!DOCTYPE html>
<html lang="es">
<head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Obligatorio Nº1 - Programación 1"/>
    <meta name="keywords" content="Universidad, Eseñanza, ORT, Tareas, Obligatorio, Programación"/>
    <meta name="author" content="Joaquín Carrasco & Gastón Barlocco" />
    <meta name="robots" content="index,follow" />

    <link rel="stylesheet" href="css/estilos.css">
    <script src="js/clases.js"> </script>
    <script src="js/funciones.js"> </script>
    <script src="https://www.google.com/jsapi"></script>

    <title>Compras compartidas</title>

</head>

<body>
<header id="inicio">
    <img id="inicio1" src="img/LogoWeb1.png" alt="Logo ">
    <img id="inicio2" src="img/LogoWeb2.png" alt="Logo ">
</header>

<nav>
    <ul>
        <li> <a href="#registro"><img src="img/perfil.png" alt="Perfil"></a> </li>
        <li> <a href="#compras"><img src="img/carrito.jpg" alt="Carro de compras"></a> </li>
        <li> <a href="#consultas"><img src="img/pregunta.jpg" alt="Ayuda" ></a> </li>
        <li> <a href="https://www.instagram.com/" target="_blank"><img src="img/instagram.svg" alt="instagram"></a> </
li>
        <li> <a href="https://www.facebook.com/" target="_blank"><img src="img/facebook.svg" alt="Facebook"></a> </li>
        <li> <a href="https://uy.linkedin.com/" target="_blank"><img src="img/linkedin.svg" alt="linkedin"></a> </li>

    </ul>
</nav>


<main>
    <section id="registro">
        <header> <h1>Personas</h1> </header>

        <article>
            <h2>Registro</h2>
```

```
            <form action="#" method="POST" id="formRegistroPersonas">

                <label for="nombrePersona"> Nombre: </label>
                <input class="redondeado" type="text" id="nombrePersona" placeholder="Nombre" maxlength="20" required
 >

                <label for="seccionPersona"> Sección: </label>
                <input class="redondeado"  type="number" id="seccionPersona" placeholder="Sección" min="1" max="8" req
uired>

                <label for="mailPersona"> Mail: </label>
                <input class="redondeado"  type="email" id="mailPersona" placeholder="Mail" required>

                <div> <button class="redondeado"  type="button" id="agregarPersona"> Agregar</button> </div>

            </form>

        </article>

        <article>
            <h2>Visualización</h2>

            <ul id="listaRegistroPersonas">
            </ul>

        </article>

        <div><a href="#inicio"> <img src="img/subir.svg" alt="Ícono inicio de página"></a></div>

    </section>

    <section id="compras">
        <header> <h1>Compras</h1> </header>

        <article>
            <h2>Registro</h2>

            <form action="#" method="POST" id="formRegistroCompras">

                <label for="registroHechaPor"> Hecha por: </label>

                <select class="redondeado" id="registroHechaPor" >
                </select>

                <label for="registroDescricpionCompra" > Descripción: </label>
                <input class="redondeado"  type="text" id="registroDescricpionCompra" placeholder="Descripción" maxlen
gth="40" required>

                <label for="registroMontoCompra" > Monto: </label>
                <input class="redondeado"  type="number" id="registroMontoCompra" placeholder="Ingrese el monto" min="
1" required>
```

```
            <p>Corresponde a:</p>
            <div id="contenedorRegistro">

            </div>

            <div> <button class="redondeado"  type="button" id="agregarCompra"> Agregar</button> </div>

        </form>

    </article>


    <article>
        <h2>Reintegro</h2>

        <form action="#" method="POST" id="formReintegroCompra">

            <label for="numeroReintegro"> Se reintegra la compra: </label>
            <select class="redondeado" id="numeroReintegro">
            </select>

            <div> <button class="redondeado" type="button" id="reintegrarCompra"> Reintegro</button> </div>

        </form>

    </article>

    <article>

        <h2>Visualización</h2>
        <label>Orden:</label>
        <label> <input id="ordenarTablaNumero" class="redondeado" type="radio" checked name="ordenVisualizacion">
número </label>
        <label> <input id="ordenarTablaNombre" class="redondeado"  type="radio" name="ordenVisualizacion"> nombre
</label>

        <table>
            <thead>
                <tr>
                    <th>NUMERO</th>
                    <th>RESPONSABLE</th>
                    <th>DESCRIPCION</th>
                    <th>MONTO</th>
                    <th>PARTICIPANTES</th>
                    <th>ESTADO</th>
                </tr>
            </thead>

            <tbody id="tablaVisualizacion">

            </tbody>
```

```
        </table>

    </article>

    <div><a href="#inicio"> <img src="img/subir.svg" alt="Ícono inicio de página" ></a></div>

</section>


<section id="consultas">
    <header> <h1>Consultas</h1> </header>

    <article>
        <h2>Pagos/Cobros pendientes por persona</h2>

        <form action="#" method="GET" id="formConsultaPersona">

            <label for="consultaSaldos"> Persona: </label>
            <select class="redondeado" id="consultaSaldos">
            </select>

            <button class="redondeado" type="button" id="consultarCompras"> Consultar totales</button>

        </form>

        <p id="participoPor"></p>
        <p id="responsablePor"></p>

    </article>


    <article>
        <h2>Descripción</h2>

        <form action="#" method="GET" id="formConsultaDescripcion">

            <label for="palabraBuscar">Palabra a buscar</label>
            <input class="redondeado" type="text" id="palabraBuscar" >
            <button class="redondeado" type="button" id="consultarDescripcion"> Consultar compras</button>

        </form>

        <h3>Resultado (se muestra primera coincidencia)</h3>

        <ul id="busquedaPalabra">

        </ul>

    </article>


    <article>
```

```
            <h2>Gráfica</h2>
            <span>Se muestra cantidad de compras totales por rangos de a $100 (0-99, 100-199, etc)</span>
            <button class="redondeado" type="button" id="graficar"> Graficar</button>

            <div id="grafico"></div>

        </article>

        <div>
            <a href="#inicio"> <img src="img/subir.svg" alt="Ícono inicio de página" ></a>
        </div>

    </section>


</main>

<footer>
    <p> Obligatorio de Programación 1 - 1er semestre 2020 - Realizado por: Joaquín Carrasco & Gastón Barlocco</p>
</footer>

</body>
</html>
```

estilos.css

```
/* Autores: Joaquín Carrasco (163609) & Gastón Barlocco (241025) */

body{
    background-color:white;
    font-family: Arial;
}

header img{
    width: 400px;
}

nav{
    background-color: blue ;
    border-radius: 50px;
}

nav ul li{
    display: inline;
    list-style: none;
}

nav ul li a img{
    width: 40px;
    padding: 10px;
    border-radius: 50%;
}

section{
    background-color: beige;
    width: 95%;
    padding-left: 5%;
    border-radius: 20px;
}

section article p{
    margin-top: 20px;
}

section h1{
    font-size: 25px;
}

section h2{
    font-size: 20px;
    margin-top: 20px;
    margin-bottom: 20px;
}

section h3{
    font-size: 15px;
```

```
    margin-top: 20px;
    margin-bottom: 20px;
}

section div a img{
    margin-top: 20px;
    width: 30px;
}

section article table thead th,td{
    border: 1px solid grey;
    text-align: center;
    padding: 5px;
}

section article form div button{
    margin-top: 20px;
}

section article ul li span{
    color: red;
    font-weight: bold;
}

#registroDescricpionCompra{
    width: 300px;
}

.redondeado{
    border-radius: 20px;
}

#contenedorRegistro{
    margin-top: 20px;
}

#grafico{
    margin-top: 2%;
    width: 800px;
    height: 600px;
}

footer{
    background-color: rgb(230, 230, 16);
    border-radius: 50px;
    font-weight: bold;
    text-align: center;
    font-size: 13px;
}

@media screen and (min-width:600px){
    #inicio1 {
```

```
        display: block;
     }
    #inicio2 {
        display: none;
     }
}

@media screen and (max-width:600px){
    #inicio1 {
        display: none;
    }
    #inicio2 {
        display: block;
        width: 250px;
        height:90px;
     }
}
```

clases.js

```
// Autores: Joaquín Carrasco (163609) & Gastón Barlocco (241025).

class Sistema {
    constructor(){
        this.listaPersonas=[];
        this.listaCompras =[];
    }

    agregarPersona(unaPersona){
        this.listaPersonas.push(unaPersona);
    }

    agregarCompra(unaCompra){
        this.listaCompras.push(unaCompra);
    }

    darTodasPersonas(){
        return this.listaPersonas;
    }

    darPersonaUnitaria(nombrePersona){
        let todasPersonas = this.darTodasPersonas();

        for (let persona of todasPersonas){
            if (persona.nombre === nombrePersona ){
                return persona;
            }
        }
    }

    darTodasComprasPorNombre(){
        this.ordenarCompraPorNombre();
        return this.listaCompras;
    }

    darTodasComprasPorNumero(){
        this.ordenarCompraPorNumero();
        return this.listaCompras;
    }

    ordenarCompraPorNombre(){
        this.listaCompras.sort(function compare(a,b){
            return a.hechoPor.nombre.localeCompare(b.hechoPor.nombre);
        });
    }

    ordenarCompraPorNumero(){
        this.listaCompras.sort(function compare(a,b){
            return a.numeroCompra - b.numeroCompra;
        });
```

```
}

darTodosNumerosCompras(){
    let listaNumerosCompras =[];
    let todasLasCompras = this.darTodasComprasPorNumero();

    for (let compra of todasLasCompras){
        if (compra.estadoCompra === "Pendiente"){
            listaNumerosCompras.push(compra.numeroCompra);
        }
    }
    return listaNumerosCompras;
}

darTodosLasDescripcionesDeBusqueda(busqueda){
    let listaDescripcionesBusqueda =[];
    let todasLasCompras = this.darTodasComprasPorNumero();

    let busquedaColorRojo = "<span>" + busqueda + "</span>";

    for (let compra of todasLasCompras){
        if (compra.descripcion.includes(busqueda) && busqueda != ""){
            listaDescripcionesBusqueda.push(compra.numeroCompra);
            listaDescripcionesBusqueda.push(compra.descripcion.replace(busqueda, busquedaColorRojo));
        }
    }
    return  listaDescripcionesBusqueda;
}

personaRepetida(nombre){
    let noEsRepetido =true;
    let todosLasPersonas =this.darTodasPersonas();

    for (let i=0; i<= todosLasPersonas.length-1 && noEsRepetido; i++ ){
        let personaUnitaria = todosLasPersonas[i];

        if ((personaUnitaria.nombre).toUpperCase() === (nombre).toUpperCase()){
            noEsRepetido = false;
        }
    }
    return noEsRepetido;
}

validacionParticipantes(responsable, participantes){
    let esCorrecto =true;

    for (let participante of participantes){
        if (participantes.length ===1){
            if (responsable.nombre === participante.nombre){
                esCorrecto = false;
            }
        }
```

```
        }
        return esCorrecto;
    }

    cambiarEstadoCompra(opcionNumeroReintegro){
        let esReintegro =false;

        let todosLasCompras = this.darTodasComprasPorNumero();

        for (let i=0; i<= todosLasCompras.length-1; i++){
            let comprasUnitarias =todosLasCompras[i];
            if(comprasUnitarias.numeroCompra == opcionNumeroReintegro){
                comprasUnitarias.estadoCompra = "Reintegrado";

                comprasUnitarias.hechoPor.responsableCompra -= comprasUnitarias.monto;

                for (let j=0; j<=comprasUnitarias.correspondeA.length-1; j++){
                    comprasUnitarias.correspondeA[j].participoCompra -
= (comprasUnitarias.monto)/(comprasUnitarias.correspondeA.length)
                }
                esReintegro =true;
            }
        }
        return esReintegro;
    }

    incrementarSaldo(nuevaCompra){
        nuevaCompra.hechoPor.responsableCompra += nuevaCompra.monto;

        for (let i=0; i<=nuevaCompra.correspondeA.length-1; i++){
            nuevaCompra.correspondeA[i].participoCompra += (nuevaCompra.monto)/(nuevaCompra.correspondeA.length);
        }
    }

    darMaximoValorCompra(){
        let maximaCompra =0;
        let todasLasCompras = this.darTodasComprasPorNombre();

        for (let compra of todasLasCompras){
            if (compra.monto >= maximaCompra){
                maximaCompra = compra.monto;
            }
        }
        return  maximaCompra;
    }

    darMontoTodasCompras(){
        let todosLosMontos =[];
        let todasLasCompras = this.darTodasComprasPorNombre();

        for (let compra of todasLasCompras){
            todosLosMontos.push(compra.monto);
```

```
        }

        return todosLosMontos;
    }

    darValoresGrafica(){
        let maxCompra=this.darMaximoValorCompra();
        let todasLasCompras = this.darMontoTodasCompras();

        let  ejeXeY = this.generarEjeXeY(maxCompra);

        let arrayGrafica = this.datosGrafico(ejeXeY,todasLasCompras);

        let datoFinal = [];
        let arrayAuxiliar=[];

        for (let i=0; i <=arrayGrafica.length-1; i++ ){
            let indiceInferior;
            let indiceSuperior;
            let cantidadCompra =arrayGrafica[i];

            indiceInferior = i*100;
            indiceSuperior = i*100 +99;

            arrayAuxiliar.push(indiceInferior+"-"+indiceSuperior);
            arrayAuxiliar.push(cantidadCompra);
            datoFinal.push(arrayAuxiliar);
            arrayAuxiliar =[];
        }
        return datoFinal;
    }


    generarEjeXeY(maximaCompra){
        /*
            Genero la estructura del eje x e y en base a la máxima compra, y lo cargo todo con 0.
            Eje x:Las posiciones serán utilizadas para realizar el indice mínimo y máximo.
            Eje y: el contenido de la posicion para la cantidad de compra.
            Ejemplo: maxima compra = 1500 --> Math.trunc(1500/100) = 15 (genero un array de 16 posiciones)
            Pos0 --> 0-99.
            Pos1 --> 100-199.
            Pos2 --> 200-299.
            .
            .
            Pos15 --> 1500-1599.
        */
        let compras =[];

        for (let i=0; i<= Math.trunc(maximaCompra/100); i++ ){
            compras.push(0);
        }
        return compras;
```

```
    }

    datosGrafico(ejeXeY,compras){
        /*
        Cargo el array respecticamente con las compras. Me pasan el eje x e y, y todas las compras.
        Cada posicion del array corresponde a un margen de compra del eje x (mínimo y máximo), entonces,
        sabiendo la compra (eje y) puedo saber su posicion e incrementar un contador. Ejemplo:
        ejeXeY =[0,0,0,0,0.....]; --> pos0: 0-99 (eje x) cant=0 (eje y). pos1: 100-199(eje x) cant=0 (eje y), etc.
        compras = [150,201,350,400......]; --> Math.trunc(150/100) =1 --> la compra 150 se ubicará
                                                                          en la posicion 1 del array ejeXeY,
                                                                          incremento ese valor.
        */

        for (let compra of compras){
            ejeXeY[Math.trunc(compra/100)] ++;
        }
        return ejeXeY;
    }
}

class Persona{
    constructor(nombre, seccion, mail){
        this.nombre =nombre;
        this.seccion =seccion;
        this.mail = mail;
        this.responsableCompra =0;
        this.participoCompra =0;
    }

    toString(){
        return this.nombre + " -Sección: " + this.seccion + "-" + this.mail;
    }
}

class Compra{
    constructor (numeroCompra, hechoPor, descripcion, monto, correspondeA){
        this.numeroCompra = numeroCompra;
        this.hechoPor = hechoPor;
        this.descripcion = descripcion;
        this.monto = monto;
        this.correspondeA =correspondeA;
        this.estadoCompra = "Pendiente";
    }

   toString(){
        return "Número de compra:" + this.numeroCompra + "Responsable: " +this.hechoPor + "Descripción: " + this.descr
ipcion + "Monto: " + this.monto;
    }
}
```

funciones.js

```
// Autores: Joaquín Carrasco (163609) & Gastón Barlocco (241025).

window.addEventListener("load",inicio);
var miSistema = new Sistema();

//Variables genericas
var conteoNumeroCompra =1;
var sePresionoConsultaCompra = false;
var sePresionoGraficar = false;


//Funcion inicio para escuchar a todos los botones de la web.
function inicio(){
    document.getElementById("agregarPersona").addEventListener("click", registrarPersonas);
    document.getElementById("agregarCompra").addEventListener("click", registrarCompras);

    document.getElementById("reintegrarCompra").addEventListener("click", reintegrarCompras);

    document.getElementById("ordenarTablaNumero").addEventListener("click", actualizarSeccionCompras);
    document.getElementById("ordenarTablaNombre").addEventListener("click", actualizarSeccionCompras);

    document.getElementById("consultarCompras").addEventListener("click", consultarCompras);
    document.getElementById("consultarDescripcion").addEventListener("click", consultarComprasDescripcion);

    document.getElementById("graficar").addEventListener("click", graficar);
}


//Funcion para registrar personas en el sistema.
function registrarPersonas(){
    let formulario = document.getElementById("formRegistroPersonas");

    if (formulario.reportValidity()){
        let nombre = document.getElementById("nombrePersona").value;
        let seccion = document.getElementById("seccionPersona").value;
        let mail = document.getElementById("mailPersona").value;

        if (miSistema.personaRepetida(nombre)){
            let nuevaPersona = new Persona (nombre, seccion, mail.toLowerCase());
            miSistema.agregarPersona(nuevaPersona);
            actualizarCargarValores();
            formulario.reset();
        }else{
            alert ("El nombre: " + nombre + " ya existe, ingrese otro nombre.")
        }
    }
}


//Funcion para actualizar carga de valores relacionados a personas.
```

```
function actualizarCargarValores(){
    agregarPersonaLista();
    cargarComboBoxNomPersonas("registroHechaPor");
    cargarComboBoxNomPersonas("consultaSaldos");
    cargarCheckBoxCorespondeA();
}


//Funcion agregar en lista de las personas registradas.
function agregarPersonaLista(){
    let lista = document.getElementById("listaRegistroPersonas");
    lista.innerHTML="";

    let todosLasPersonas = miSistema.darTodasPersonas();

    for (let personaUnitaria of todosLasPersonas){
        agregarLista("listaRegistroPersonas", personaUnitaria);
    }
}


//Funcion para cargar en todos los comboBox de la web el nombre de las personas (Hecha por, busqueda).
function cargarComboBoxNomPersonas(id){
    let select = document.getElementById(id);
    select.innerHTML="";

    let todosLasPersonas = miSistema.darTodasPersonas();

    for (let persona of todosLasPersonas ){
        agregarOption(id, persona.nombre);
    }
}


//Funcion para cargar el checkBox con los nombres de las personas registradas en el sistema.
function cargarCheckBoxCorespondeA(){
    let contenedorCorrespondeA = document.getElementById("contenedorRegistro");
    contenedorCorrespondeA.innerHTML="";

    let todosLasPersonas = miSistema.darTodasPersonas();

    for (let persona of todosLasPersonas){
        let nodoDiv = document.createElement("div");
        let nodoLabel = document.createElement("label");
        let nodoSpan = document.createElement("span");
        let nodoText = document.createTextNode(persona.nombre);

        let nodoInput = document.createElement("input");
        nodoInput.setAttribute("type", "checkbox");
        nodoInput.setAttribute("name", "nombreCheck");
        nodoInput.setAttribute("value", persona.nombre);
```

```
        nodoSpan.appendChild(nodoText);
        nodoLabel.appendChild(nodoInput);
        nodoLabel.appendChild(nodoSpan);
        nodoDiv.appendChild(nodoLabel);
        contenedorCorrespondeA.appendChild(nodoDiv);

    }
}


//Funcion para registrar compras en el sistema.
function registrarCompras(){
    let formulario = document.getElementById("formRegistroCompras");

    if (formulario.reportValidity()){
        let numeroItemCompra =conteoNumeroCompra;

        let selectComprador = document.getElementById("registroHechaPor");
        let opcionComprador= selectComprador.options[selectComprador.selectedIndex].value;

        let registroDescripcion = document.getElementById("registroDescricpionCompra").value;
        let registroMontoCompra = parseInt(document.getElementById("registroMontoCompra").value);
        let correspondeA = selectCheckCorrespondeA();

        let responsableCompra = miSistema.darPersonaUnitaria(opcionComprador);

        if(miSistema.validacionParticipantes(responsableCompra,correspondeA) && correspondeA.length !=0 ){
            conteoNumeroCompra++;

            let nuevaCompra = new Compra (numeroItemCompra, responsableCompra ,registroDescripcion,registroMontoCompra
, correspondeA );
            miSistema.agregarCompra(nuevaCompra);
            miSistema.incrementarSaldo(nuevaCompra);
            actualizarSeccionCompras();
            formulario.reset();
        }
        if(!(miSistema.validacionParticipantes(responsableCompra,correspondeA))){
            alert("Error: el responsable no puede ser el único participante. ");
        }
        if(correspondeA.length == 0 ){
            alert("Error: debe seleccionar algun participante de la compra. ");
        }
    }
}


//Funcion para retornar las personas seleccionadas en el checkBox.
function selectCheckCorrespondeA(){
    let items=document.getElementsByName("nombreCheck")
    let seleccionItems=[];
    let persona ="";
```

```
    for(let i=0; i<= items.length-1; i++){
        if(items[i].checked==true){
            persona=items[i].value;
            seleccionItems.push(miSistema.darPersonaUnitaria(persona));
        }
    }
    return(seleccionItems);
}


//Funcion para actualizar seccion compras (tabla, combobox).
function actualizarSeccionCompras(){
    agregarTabla();
    agregarReintegroComboBox();
    graficarValores();
}


//Funcion para agregar todas las compras a la tabla.
function agregarTabla(){
    let tabla = document.getElementById("tablaVisualizacion");
    tabla.innerHTML="";

    let todasLasCompras;

    if (document.getElementById("ordenarTablaNombre").checked){
        todasLasCompras= miSistema.darTodasComprasPorNombre();
    }else{
        todasLasCompras=miSistema.darTodasComprasPorNumero();
    }

    for (let compra of todasLasCompras) {
        let fila =tabla.insertRow();
        let celda = fila.insertCell();
        celda.innerHTML = compra.numeroCompra;

        celda = fila.insertCell();
        celda.innerHTML = compra.hechoPor.nombre;

        celda = fila.insertCell();
        celda.innerHTML = compra.descripcion;

        celda = fila.insertCell();
        celda.innerHTML = compra.monto;

        let participantesCompra =[];

        for (let persona of compra.correspondeA){
            participantesCompra.push(persona.nombre);
        }
        celda = fila.insertCell();
        celda.innerHTML = participantesCompra;
```

```
        celda = fila.insertCell();
        celda.innerHTML = compra.estadoCompra;
    }
}


//Funcion que agrega el número de compra en el comboBox de reintegro.
function agregarReintegroComboBox(){
    let numerosCompra = miSistema.darTodosNumerosCompras();
    let select = document.getElementById("numeroReintegro");
    select.innerHTML="";

    for (let numero of numerosCompra ){
        agregarOption("numeroReintegro", numero);
    }
}


//Funcion para reintegrar compras en el sistema.
function reintegrarCompras(){
    let selectNumeroReintegro= document.getElementById("numeroReintegro");
    let opcionNumeroReintegro= selectNumeroReintegro.options[selectNumeroReintegro.selectedIndex].value;

    if (miSistema.cambiarEstadoCompra(opcionNumeroReintegro)){
        actualizarSeccionCompras();
    }
    consultarComprasPersona();
}


//Funcion para consultar las compras en base a la descripcion ingresada.
function consultarComprasDescripcion(){
    let descripcionBusqueda = document.getElementById("palabraBuscar").value;
    let resultadonBusqueda = miSistema.darTodosLasDescripcionesDeBusqueda(descripcionBusqueda);
    // resultadonBusqueda = [numeroDeCompra,descripcion,numeroDeCompra,descripcion,etc].

    let lista = document.getElementById("busquedaPalabra");
    lista.innerHTML="";

    for (let i=0; i<resultadonBusqueda.length; i+=2 ){
        lista.innerHTML += "<li>" +"Compra " + resultadonBusqueda[i] + " " + resultadonBusqueda[i+1] + "</li>";
    }
}


//Funcion para consultar compras.
function consultarCompras(){
    sePresionoConsultaCompra = true;
    consultarComprasPersona();
}
```

```
//Funcion para consultar compras de personas.
function consultarComprasPersona(){

    if (sePresionoConsultaCompra){
        let selectPersona = document.getElementById("consultaSaldos");
        let opcionPersona= selectPersona.options[selectPersona.selectedIndex].value;

        let persona = miSistema.darPersonaUnitaria(opcionPersona);

        let participoEnCompras = persona.participoCompra.toFixed(2);
        let responsableDeCompra = persona.responsableCompra.toFixed(2);

        let nodoParrafo = document.getElementById("participoPor");
        nodoParrafo.innerHTML="";

        let contenidoParrafo = "Participó en total por $" + participoEnCompras;
        agregarParrafo("participoPor",contenidoParrafo);

        nodoParrafo = document.getElementById("responsablePor");
        nodoParrafo.innerHTML="";

        contenidoParrafo = "Responsable de compras por $" + responsableDeCompra;
        agregarParrafo("responsablePor",contenidoParrafo);
    }
}


//Funcion para graficar.
function graficar(){
    sePresionoGraficar = true;
    graficarValores();
}


//Funcion para graficar valores.
function graficarValores(){
    if (sePresionoGraficar){
        google.load("visualization", "1", {packages:["corechart"]});

        google.setOnLoadCallback(dibujarGrafico);

        function dibujarGrafico() {
            let datosGrafica = miSistema.darValoresGrafica();

            var data = new google.visualization.DataTable();
            data.addColumn('string', 'Margen de compras');
            data.addColumn('number', 'Cantidad de compras');


            data.addRows(datosGrafica);

            var options = { "title": "Rangos",
```

```
            vAxis: {
                    format: "0",
                }
            };

            var grafica = new google.visualization.ColumnChart(document.getElementById('grafico'));
            grafica.draw(data, options);
        }
    }
}


//Funciones genericas

//Crear lista, crear opcion en select
function agregarLista(id, texto){
    let lista = document.getElementById(id);
    let nodoLi =document.createElement("li");
    let nodoText = document.createTextNode(texto);

    nodoLi.appendChild(nodoText);
    lista.appendChild(nodoLi);
}


//Agregar option en select. Le pasamos a la funcion el ID del select + el texto a ingresa.
function agregarOption(id, texto){
    let select = document.getElementById(id);
    let nodoOption =document.createElement("option");
    let nodoText = document.createTextNode(texto);

    nodoOption.appendChild(nodoText);
    select.appendChild(nodoOption);
}


function agregarParrafo(id,texto){
    let nodoParrafo = document.getElementById(id);
    let nodoText = document.createTextNode(texto);

    nodoParrafo.appendChild(nodoText);
}
```