

Universidad ORT Uruguay
Facultad de Ingeniería

Documentación para el Obligatorio 2

Entregado como requisito para la obtención del puntaje
para la materia Programación

Joaquín Bonifacino - 243193
Rafael Cadenas – 269150

Tutores: Gonzalo Esteban Wagner Lemos y Sebastián
Andrés Pesce Muñoz

2021

Declaración de Autoría:

Nosotros, Joaquín Bonifacino y Rafael Cadenas, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el obligatorio numero 2 julio 2021
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad.
- En la obra, hemos acusado recibo de las ayudas recibidas
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros.
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Firma de autor

Joaquín Bonifacino
Aclaración de firma



Firma de autor

Rafael Cadenas
Aclaración de firma

Código

HTML:

```
<!--Joaquin Bonifacino y Rafael Cadenas-->

<!doctype html>

<html lang="es">

    <head>

        <link rel="stylesheet" href="css/styles.css">

        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-
scale=1.0">

        <meta name="description" content="Pagina de Donaciones para una
Institucion">

        <script src="js/clases.js"></script>

        <script src="js/script.js"></script>

        <script src="https://www.gstatic.com/charts/loader.js"></script>

        <title>Sistema de Donaciones</title>

    </head>

    <body>

        <nav>

            <h2 id="tituloNav">Sistema de donaciones</h2>

            

            <a class="menuizq1" href="#donantes">Donantes</a>

            <a class="menuizq1" href="#donaciones">Donaciones</a>

            <a class="menuizq1" href="#estadisticas">Estadísticas</a>

            <em id="total">Total General <br> sin datos</em>

            <em id="mayor">Monto Donación Mayor <br> sin datos</em>

        </nav>

    <section>
```

```

<h2 id="tituloSect"></h2>

<aside id="donantes" class="cont-sect">

    <h2 class="titulos">Registro de Donantes</h2>

    <form method="get" action="#" id="idFormDonantes">

        <label for="nombre">Nombre</label>

        <input type="text" id="nombre" name="nombre"
placeholder="Nombre" maxlength="20" required="">

        <label for="direccion">Direccion</label>

        <input type="text" id="direccion"
name="direccion" placeholder="Dirección" maxlength="20" required="">

        <label for="telefono">Telefono</label>

        <input type="text" id="telefono" name="telefono"
placeholder="Teléfono" maxlength="20" required="">

        <input id="infoDonantes" type="button"
value="Agregar">

    </form>

</aside>

<aside id="donaciones" class="cont-sect">

    <form method="get" action="#" id="idFormDonaciones">

        <h2 class="titulos">Registro de Donaciones</h2>

        <label for="idDonante">Donantes</label>

        <select name="Donante" id="idDonante">

            </select>

        <label for="idModo">Modo</label>

        <select name="Modo" id="idModo">

            <option value="Efectivo"
selected>Efectivo</option>

```

```

        <option
value="Transferencia">Transferencia</option>
        <option value="Canje">Canje</option>
        <option
value="Mercaderia">Mercaderia</option>
        <option value="Cheque">Cheque</option>
        <option value="Otros">Otros</option>
    </select>

    <label for="idMonto">Monto</label>
    <input type="number" id="idMonto" name="Monto"
min="1" required="">

    <label for="idComentarios">Comentarios</label>
    <input type="text" id="idComentarios"
name="Comentarios">

    <input id="infoDonaciones" type="button"
value="Agregar">

    <br>
</form>
<h3>Donaciones Registradas</h3>
<div id="Orden">
    <label>Orden:</label>
    <input type="radio" id="idDecreciente" name="Orden"
checked="">
    <label for="idDecreciente">Monto Decreciente</label>
    <input type="radio" id="idCreciente" name="Orden">
    <label for="idCreciente">Nombre donante creciente</label>
</div>
<table id="tablaDonaciones">

```

```

</table>

<p>Resaltar filas:</p>
<input type="checkbox" id="idResaltar" checked>
<p>de monto:</p>
<input type="number" min="0" id="idBuscar" name="Buscar"
placeholder="0">
</aside>
<aside id="estadisticas" class="cont-sect">
  <h2 class="titulos">Estadísticas</h2>
  <p id="idTotalDonaciones">Cantidad total de donaciones: 0</p>
  <p id="idPromedioMonto">Promedio por donación: sin
datos</p>
  <p id="texto">Donante que más veces donó: sin datos</p>
  <ul id="idDonanteMayor">

  </ul>
  <div id="donutchart"></div>
</aside>
</section>
<footer>Cátedra de programación - Programación 1 - 2021 - Joaquin Bonifacino
/ 243193 - Rafael Cadenas / 269150</footer>
</body>

</html>

```

CSS:

```
/*Joaquin Bonifacino y Rafael Cadenas-->*/
```

```
    @import  
    url('https://fonts.googleapis.com/css2?family=Roboto:wght@100&display=swap');
```

```
    body{  
        font-family: 'Roboto', sans-serif;  
        background-color: #d3d3d3;  
        font-weight: bold;  
    }
```

```
    #Logo{  
        border-radius: 100%;  
        width: 40%;  
        height: 40%;  
        background-color: #f3f3f3;  
        margin-left: 25%;  
        margin-top: 5%;  
        margin-bottom: 5%;  
    }
```

```
    .menuizq1{  
        background-color: #72a2cf;  
        display: block;  
        margin-left: 1%;  
        width: 90%;  
        border-radius: 10px;
```

```
margin-top: 1%;
padding: 0.5%;
text-align: center;
border: 2px solid #0015af;
font-size: 140%;

}

#tituloSect{
    display: none;
}

#tituloNav{
    text-align: center;
}

section{
    background-color:blue;
    float: right;
    width: 78%;
    background-color: #bababa;
    padding-left: 2%;
    padding-bottom: 8.95%;

}

nav{
    float:left;
    background-color: #d3d3d3;
```



```
width: 20%;  
}
```

```
footer{  
    background-color: #6d6d6d;  
    text-align: center;  
    color: white;  
    padding-bottom: 1%;  
    padding-top: 1%;  
    font-size: 14px;  
    clear:both;  
    width: 100%;  
}
```

```
#total{  
    background-color: #3a6fa2;  
    display:block;  
    margin-left: 1%;  
    width: 90%;  
    border-radius: 30px;  
    margin-top: 10%;  
    padding: 0.5%;  
    text-align: center;  
    color: white;  
    border-style: double;  
    font-size: 150%;  
}
```

```
#mayor{
```

```
background-color: #3a6fa2;
display: block;
margin-left: 1%;
width: 90%;
border-radius: 30px;
margin-top: 2%;
padding: 0.5%;
text-align: center;
color: white;
border-style: double;
font-size: 150%;

}
```

```
.cont-sect{
    font-size: 20px;
    padding-left: 7%;
}
```

```
.titulos{
    font-size: 35px;

}
```

```
body{
    margin: 0px;
    overflow: auto;
```

```
}
```

```
nav h2{  
    margin:0px;  
    padding: 1%;  
    font-size: 32px;  
}
```

```
#idMonto{  
width: 4%;  
}
```

```
#idComentarios{  
    width: 15%;  
}
```

```
table{  
    width:90%;  
    text-align: center;  
    padding-top:  
}
```

```
table, th, td {  
    border: 1px solid black;  
}
```

```
#Orden{  
    padding-bottom: 1.5%  
}
```

```
#idEfectivo{  
    color:red;  
}
```

```
.noEfectivo{  
    color: green;  
}
```

```
#donaciones p{  
    display: inline-block;  
}
```

```
#donaciones h2{  
    padding-top: 2%;  
}
```

```
#donaciones h3{  
    padding-top: 1%;  
}
```

```
ul{  
    list-style-image: url("../img/icons8-pica-24.png")  
}
```

```
@media screen and (max-width: 700px) {  
    section{
```

```
float: none;

width: 100%;
}

body{

width: 100%;

}

nav{

float: none;

width: 95%;
}

.menuizq1{

width: 100%;
}

nav h2{

text-align: center;

margin:0px;

padding: 1%;

font-size: 32px;

padding-top: 3%;
}

#Logo{

margin-left: 30%;
}

#total{
```

```
        width: 100%;  
    }  
    #mayor{  
        width: 100%;  
    }  
}
```

Script.js:

```
/*Joaquin Bonifacino y Rafael Cadenas*/
```

```
//Evento que se ejecuta al terminar de cargar la pagina
```

```
window.addEventListener("load",inicio);
```

```
let nuestroSistema = new Sistema();
```

```
function inicio(){
```

```
    //Desabilita el boton de agregar una donacion
```

```
    document.getElementById("infoDonaciones").disabled = true;
```

```
    //Trigger de agregar Donaciones
```

```
    document.getElementById("infoDonantes").addEventListener("click",agregarDonante);
```

```
    //Trigger de agregar Donantes
```

```
    document.getElementById("infoDonaciones").addEventListener("click",agregarDonacion);
```

```
    //Trigger de ordenar por creciente o decreciente
```

```
    document.getElementById("idDecreciente").addEventListener("change",actualizar);
```

```
    document.getElementById("idCreciente").addEventListener("change",actualizar);
```

```
    //Trigger de ordenar por creciente o decreciente
```

```
    document.getElementById("idResaltar").addEventListener("change",actualizar);
```

```
    document.getElementById("idBuscar").addEventListener("input",actualizar);
```

```
    //Carga los paquetes de graficas que ofrece google
```

```
    google.charts.load("current", {packages:["corechart"]});
```

```
    google.charts.setOnLoadCallback(drawChart);
```

```

//Carga la tabla (Aunque no contenga datos)
    nuestroSistema.actualizarTabla();
}

function agregarDonante(){

    if(document.getElementById("idFormDonantes").reportValidity()){

        //Traemos las 3 variables de los 3 inputs de la pagina
        let nombre = document.getElementById("nombre").value;
        let direccion = document.getElementById("direccion").value;
        let tel = document.getElementById("telefono").value;

        //En el caso de que el donante no exista se agrega, sino se alerta
        if(!nuestroSistema.existeDonanteNombre(nombre)){
            //Creamos el nuevo objeto donante y le pasamos las variables
correspondientes
            let don = new Donante(nombre,direccion,tel);

            //Agregamos el objeto al array de donantes
            nuestroSistema.agregarDonanteLista(don);

            //Actualizamos el combo de donaciones con el nuevo donante
            actualizarCombo();

            //Limpiamos los campos de input una vez se agrego el donante
            document.getElementById("idFormDonantes").reset();

            //Rehabilita el boton de donaciones
            document.getElementById("infoDonaciones").disabled = false;

```



```

        alert("Donante registrado con exito");
    }else{
        alert("Este donante ya existe");
    }

    }else{
        alert("Nombre, direccion o telefono incorrectos")
    }
}

```

```

function actualizarCombo(){

    /*Traemos el array de todos los donantes y el objeto del combo,
    luego limpiamos lo que tenga dentro el combo
    */

    let listaDonantes = nuestroSistema.darTodosDonantes();
    let combo = document.getElementById("idDonante");
    combo.innerHTML = "";

    //Recorriendo el array de donantes re cargamos el combo con los nombres
    for(let i=0; i<listaDonantes.length; i++){

        let donante = listaDonantes[i];
        let nodoOption = document.createElement("option");
        let nodoTxt = document.createTextNode(donante.nombre);
    }
}

```

```

//Despues de cargadas las variables usamos append para cargar el combo
nodoOption.appendChild(nodoTxt);
combo.appendChild(nodoOption);
}
}

function agregarDonacion(){

    //Chequea si todos los campos han sido correctamente ingresados
    if(document.getElementById("idFormDonaciones").reportValidity()){

        //Traigo los valores de cada uno de los inputs
        let combo = document.getElementById("idDonante");
        let nombre = combo.options[combo.selectedIndex].text;
        let modo = document.getElementById("idModo").value;
        let monto = parseInt(document.getElementById("idMonto").value);
        let comentario = document.getElementById("idComentarios").value;

        // Creamos el objeto donacion con los valores adecuados a la clase
        let donacion = new Donacion(nombre,modo,monto,comentario);

        //Se agrega el objeto al array de donaciones
        nuestroSistema.agregarDonacionLista(donacion);

        //Funcion que actualiza la lista de donante mayor
        actualizarDonanteMayor();

        //Convierte el atributo nombre del objeto donacion en el objeto
        donante en si
    }
}

```

```

nuestroSistema.agregarDonanteADonacion();

//Actualiza la tabla
nuestroSistema.actualizarTabla();
alert("Donacion registrada con exito");

//Actualiza el Monto total General
actualizarTotalMonto();

//Actualiza en estadisticas el monto de donacion mayor
actualizarMayor();

//Actualiza en estadisticas la cantidad de donaciones
actualizarCantTDonaciones();

//Actualiza en estadisticas el promedio de los montos de las donaciones
actualizarPromedioMonto();

//Re dibuja el chart despues de agregada una donacion
drawChart();

}else{
    alert("Nombre, modo, monto o comentario incorrectos")
}
}

//Actualiza el Monto total General
function actualizarTotalMonto(){
    /*Guarda el objeto en "total", acumula en la variable "suma"

```

```

    luego lo concatena al texto y lo re carga al objeto
    */
    let total = document.getElementById("total");
    let suma = nuestroSistema.total();
    suma = "Total General $" + suma;
    total.innerHTML = suma;
}

//Actualiza en estadísticas el monto de donacion mayor
function actualizarMayor() {
    let mayor = document.getElementById("mayor");
    let valor = nuestroSistema.mayor();
    valor = "Monto Donación Mayor $" + valor;
    mayor.innerHTML = valor;
}

//Actualiza la cantidad de donaciones totales mostrada en la seccion de estadísticas
function actualizarCantTDonaciones(){
    let objeto = document.getElementById("idTotalDonaciones");
    let cantidad = nuestroSistema.listaDonaciones.length;
    let string = "Cantidad total de donaciones:" + cantidad;
    objeto.innerHTML = string;
}

//Actualiza el promedio de monto de las donaciones mostrado en la seccion de estadísticas
function actualizarPromedioMonto(){
    let objeto = document.getElementById("idPromedioMonto");
    let listaDonaciones = nuestroSistema.darTodosDonacion();
    let suma = 0;
    for (let i = 0; i < listaDonaciones.length; i++) {

```

```

        suma+=parseInt(listaDonaciones[i].monto);
    }
    let promedio = suma/listaDonaciones.length;
    promedio = parseInt(promedio);
    let string = "Promedio por donación: " + promedio;
    objeto.innerHTML = string;
}

function actualizarDonanteMayor(){
    let mayor = document.getElementById("idDonanteMayor");
    mayor.innerHTML = "";
    let donantes = nuestroSistema.mayorDonante();
    let p = document.getElementById("texto");
    //Se valida si es uno o mas donantes para determinar el singular o plural
    if(donantes.length > 1){
        p.innerHTML = "Donantes que más veces donaron:";
        //Se recorre la lista de los donantes que mas donaron y se agregan a la
        lista en el HTML
        for (i = 0; i < donantes.length; i++){
            texto = donantes[i];
            let donante = document.createElement("li");
            let txt = document.createTextNode(texto);
            donante.appendChild(txt);
            mayor.appendChild(donante);
        }
    } else {
        let txt = "Donante que más veces donó: " + donantes;
        p.innerHTML = txt;
    }
}

```

```

}

//creada para que funcione con los event listeners
function actualizar(){
    nuestroSistema.actualizarTabla();
}

//Funcion que dibuja la grafica
function drawChart() {
    let cantEfectivo=0;
    let cantTrans=0;
    let cantCanj=0;
    let cantMerc=0;
    let cantCheque=0;
    let cantOtros=0;

    //Recorre la lista de donaciones comprobando que modo usa
    for (var i = 0; i < nuestroSistema.listaDonaciones.length; i++) {
        if(nuestroSistema.listaDonaciones[i].modo=="Efectivo"){
            cantEfectivo++;
        }

        if(nuestroSistema.listaDonaciones[i].modo=="Transferencia"){
            cantTrans++;
        }

        if(nuestroSistema.listaDonaciones[i].modo=="Canje"){
            cantCanj++;
        }

        if(nuestroSistema.listaDonaciones[i].modo=="Mercaderia"){

```

```

        cantMerc++;
    }

    if(nuestroSistema.listaDonaciones[i].modo=="Cheque"){
        cantCheque++;
    }

    if(nuestroSistema.listaDonaciones[i].modo=="Otros"){
        cantOtros++;
    }

}

//Se establecen las variables que se toman en cuenta en la grafica
var data = google.visualization.arrayToDataTable([
    ['Modo', 'Cantidad'],
    ['Efectivo',cantEfectivo],
    ['Transferencia',cantTrans],
    ['Canje',cantCanj]],
    ['Mercaderia',cantMerc],
    ['Cheque',cantCheque],
    ['Otros',cantOtros]
]);

    //Valida el titulo dependiendo si hay datos o no
    let title;
    title= 'Donaciones por Modo';

    //Establece el diseño
    var options = {

```

```
        title: title,
        pieHole: 0.4,
        backgroundColor:"none",
    };

    //Crea el objeto de la clase piechart

    var chart = new
    google.visualization.PieChart(document.getElementById('donutchart'));

    if (nuestroSistema.listaDonaciones.length > 0){
        document.getElementById("donutchart").style.width = "900px";
        document.getElementById("donutchart").style.height= "500px";
        chart.draw(data, options);
    }
}
```


clases.js:

```
class Sistema{  
  constructor(){  
    this.listaDonantes = [];  
    this.listaDonaciones = [];  
  }  
  
  //Devuelve la lista de todos los donantes  
  darTodosDonantes(){  
    return this.listaDonantes;  
  }  
  
  //Devuelve la lista de todas las donaciones  
  darTodosDonacion(){  
    return this.listaDonaciones;  
  }  
  
  //Convierte el atributo nombre del objeto donacion en el objeto donante en si  
  agregarDonanteADonacion(){  
    let listaDonantes = this.darTodosDonantes();  
    let listaDonaciones = this.darTodosDonacion();  
    for (let i2 = 0; i2 < listaDonantes.length; i2++) {  
      let nombre = listaDonantes[i2].nombre  
  
      for (let i = 0; i < listaDonaciones.length; i++) {  
        if(listaDonaciones[i].donante == nombre)  
          listaDonaciones[i].donante = listaDonantes[i2];  
      }  
    }  
  }  
}
```

```
}
```

```
volverANombre(){  
    let listaDonantes = this.darTodosDonantes();  
    let listaDonaciones = this.darTodosDonacion();  
    for (let i2 = 0; i2 < listaDonantes.length; i2++) {  
        let nombre = listaDonantes[i2].nombre  
  
        for (let i = 0; i < listaDonaciones.length; i++) {  
            if(listaDonaciones[i].donante.nombre == nombre)  
                listaDonaciones[i].donante = nombre;  
        }  
    }  
}
```

```
//Funcion que se le apsa un objeto para agregar al array de donantes
```

```
agregarDonanteLista(unElemento){  
    this.listaDonantes.push(unElemento);  
}
```

```
//Funcion que se le apsa un objeto para agregar al array de donaciones
```

```
agregarDonacionLista(unElemento){  
    this.listaDonaciones.push(unElemento);  
}
```

```
//Funcion que devuelve verdadero en el caso de que ya exista un donante con ese  
nombre
```

```
//(ignorando mayusculas y minusculas)
```

```

existeDonanteNombre(nombre){

    let existe = false;

    for(let i=0; i<this.listaDonantes.length; i++){

        let d = this.listaDonantes[i];

        if(d.nombre.toLowerCase() == nombre.toLowerCase()){
            existe = true;
        }

    }

    return existe;
}

/*Funcion encargada de tomar cada monto del sistema, parsearlo
y acumularlo en la variable "suma" que es retornada
*/
total(){
    let suma = 0;
    let listaDonaciones = this.darTodosDonacion();
    for (let i = 0; i < listaDonaciones.length; i++) {
        let d = listaDonaciones[i];
        suma = suma + parseInt(d.monto)
    }
    return suma
}

```

```
}
```

```
/*Funcion encargada de buscar la mayor donacion dentro del array de donaciones  
y devolverla en la variable "max"
```

```
*/
```

```
mayor(){  
    let max = Number.MIN_VALUE;  
    let listaDonaciones = this.darTodosDonacion();  
    for (let i = 0; i < listaDonaciones.length; i++){  
        let d = listaDonaciones[i];  
        let m = parseInt(d.monto);  
        if(m > max) {  
            max = m;  
        }  
    }  
    return max;  
}
```

```
//Funcion encargada de devolver un array el cual trae el/los donante/s con el mayor  
monto
```

```
mayorDonante(){  
    this.volverANombre();  
    let listaDonantes = this.darTodosDonantes();  
    let listaDonaciones = this.darTodosDonacion();  
    let max = Number.MIN_VALUE;  
    let donantes = [];  
  
    for (let i2 = 0; i2 < listaDonantes.length; i2++) {  
        let nombre = listaDonantes[i2].nombre;
```

```

let cont = 0;

for (let i = 0; i < listaDonaciones.length; i++) {
    if(listaDonaciones[i].donante == nombre){
        cont++;
    }
}

if(cont > max){
    max = cont;
    donantes = [];
}

if(cont == max){
    nombre = "" + nombre
    donantes.push(nombre);
}

}

return donantes;
}

```

//Funcion que ordena la tabla chequeando que radio button esta apretado cuando es llamada

```

ordenarLista(){
    let radio = document.getElementById("idDecreciente");
    if(radio.checked){
        this.listaDonaciones.sort(function(a,b){

```

```

        return b.monto - a.monto;
    });
} else {
    this.listaDonaciones.sort(function(a,b){
        return a.donante.toString().localeCompare(b.donante.toString())
    });
}
}
}

```

//Funcion que se le pasa un objeto combo de argumento y devuelve si esta chequeado o no

```

comboChequeado(combo){
    let check=false;
    if(combo.checked){
        check=true
    }
    return check;
}

```

//Funcion que se encarga de actualizar la tabla

```

actualizarTabla(){

```

```

    let tabla = document.getElementById("tablaDonaciones");

```

```

    let listaDonaciones = this.darTodosDonacion();

```

```

    let listaDonantes = this.darTodosDonantes();

```

//Si no hay donaciones ingresada se muestra el mensaje "No hay donaciones para mostrar"

```

    if(listaDonaciones.length == 0){

```

```

let fila = tabla.insertRow();
let celda = fila.insertCell();
celda.colSpan="4";
celda.innerHTML = "No hay donaciones para mostrar";

}else{

//Limpio la tabla original
tabla.innerHTML = "";

//Cargo cada uno de los headers de la tabla
//Header 1
var header = tabla.createTHead();
var row = header.insertRow(0);
var cell = row.insertCell(0);
cell.innerHTML = "<b>Comentarios</b>";
//Header 2
header = tabla.createTHead();
cell = row.insertCell(0);
cell.innerHTML = "<b>Monto</b>";
//Header 3
header = tabla.createTHead();
cell = row.insertCell(0);
cell.innerHTML = "<b>Modo</b>";
//Header 4
header = tabla.createTHead();
cell = row.insertCell(0);
cell.innerHTML = "<b>Donante</b>";

```

```

//Llamo a la funcion que orden por monto o nombre
this.ordenarLista();

//Traigo el valor del monto a resaltar
let checkbox = document.getElementById("idResaltar");
if(nuestroSistema.comboChequeado(checkbox)){
    var montoAResaltar= document.getElementById("idBuscar").value;

}

/*Recorro la tabla de donaciones para cargar la tabla
en el caso de que el valor a resaltar sea igual a algun monto de cualquier
donaciones
la fila que le corresponde se le cambia el estilo del fondo a amarillo.
*/
for(let i=0; i<listaDonaciones.length; i++){

    let fila = tabla.insertRow();

    if(montoAResaltar==listaDonaciones[i].monto){
        fila.style.backgroundColor = "yellow";
    }

    let celda = fila.insertCell();

    //Se carga la correspondiente informacion del donante y el modo.
    celda.innerHTML = listaDonaciones[i].donante.toString();

    let celda2 = fila.insertCell();
    celda2.innerHTML = listaDonaciones[i].modo;

```



```

    /*Se crea la celda para el monto y despues se cambia el estilo del color del
texto
de esa celda dependiendo de si el monto es mayor o menor a 1000
*/
    let celda3 = fila.insertCell();
    if(listaDonaciones[i].monto >= 1000){
        celda3.style.color = "red";
        celda3.innerHTML = listaDonaciones[i].monto;
    } else {
        celda3.style.color = "green";
        celda3.innerHTML = listaDonaciones[i].monto;
    }
    let celda4 = fila.insertCell();
    celda4.innerHTML = listaDonaciones[i].comentario;

}
}
}
}

```

```

//Creacion de la clase donante con sus 3 respectivos atributos
class Donante{
    constructor(nombre,direccion,telefono){
        this.nombre= nombre;
        this.direccion = direccion;
    }
}

```

```

        this.telefono = telefono;
    }

    //Funcion que devuelve la informacion del objeto donante desde la que fue llamada
    toString(){
        return this.nombre +" ("+ this.direccion+ "," +this.telefono+")" ;
    }

}

//Creacion de la clase donante con sus 4 respectivos atributos
class Donacion{
    constructor(donante,modo,monto,comentario){
        this.donante = donante;
        this.modo = modo;
        this.monto = monto;
        this.comentario = comentario;
    }

    //Devuelve las propiedades del objeto donde es llamado
    toString(){
        return this.donante + "/" + this.modo+ "/" + this.monto + "/" +this.comentario ;
    }

}

```