

MEMORIA

SAN - JE SHOP

Tienda N°1 en España de comida rápida asiática

Autores
Samuel Hernández Pérez
Jose Raúl Herrero Alarcón

Base de Datos.....	3
Modelo relacional.....	3
Modelo Entidad/Relación.....	4
Esquema.....	4
Explicación.....	5
PROGRAMACIÓN.....	7
Servlet.....	7
JSP.....	15
Lenguaje de marcas.....	24
Mockup.....	24
Factura en XML para conversión a PDF.....	40
Conclusiones y dificultades encontradas.....	41
Posibles mejoras o ampliaciones.....	42
Bibliografía.....	43

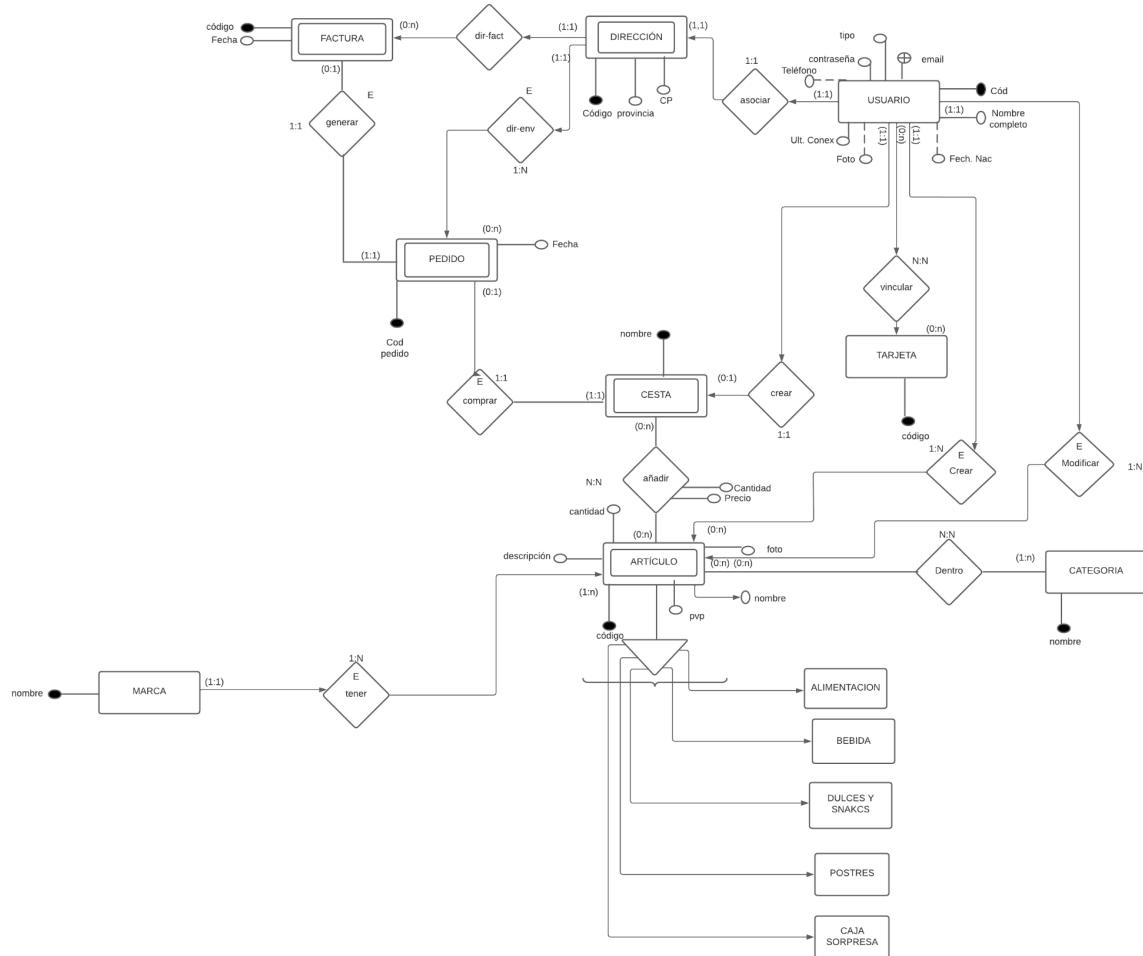
Base de Datos

Modelo relacional

<p>FACTURA (<u>código</u>, Fecha, Dir_Facturación, Cód_dir) PK: código VNN: Cód_dir UK: Cód_pedido FK: Cód_dir → DIRECCIÓN.Código FK: Cód_pedido → PEDIDO.Código</p>	<p>DIRECCIÓN (<u>Código</u>, provincia, CP, tipo, dirección, Cód_usu) PK: Código VNN: Cód_usu FK: Cód_usu → USUARIO.Cód_usu</p>
<p>USUARIO (<u>Cód</u>, Nombre completo, Fech. Nac, Foto, Ult. Conex, Teléfono, Contraseña, email) PK: Cód VNN: email UK: email</p>	<p>PEDIDO (<u>Código</u>, Fecha, Precio, Cantidad, nom_cesta,cod_usu, Cód_dir) PK: Código UK: nom_cesta,cod_usu VNN: Cód_dir FK: nom_cesta, cod_usu → CESTA.(nombre,cod_usu) FK: Cód_dir → DIRECCIÓN.Código</p>
	<p>VINCULAR_TARJETA (<u>Cód_usu,Cód_tarj</u>) PK: Cód_usu,Cód_tarj FK: Cód_usu → USUARIO.Cód_usu FK: Cód_tarj → TARJETA.código</p>
<p>TARJETA (código) PK: código</p>	<p>CESTA (<u>nombre, cód_usu, tipo</u>) PK: nombre, cód_usu FK: cód_usu → USUARIO.cód_usu</p>
<p>ARTICULO_A_CESTA (<u>código_art,nom_cesta,cód_usu</u>, Precio, Cantidad) PK: código_art,nom_cesta,cod_usu FK: código_art → ARTÍCULO.código FK: nom_cesta, cod_usu → CESTA.(nombre,cod_usu)</p>	<p>ARTÍCULO (<u>código</u>, pvp, nombre, descripción, foto, cantidad, nombre_marca, Cód_usu) PK: código VNN: Cód_usu VNN: nombre_marca FK: nombre_marca → MARCA.nombre FK: Cód_usu → USUARIO.Cód_usu</p>
<p>CATEGORÍA (<u>nombre</u>) PK: nombre</p>	<p>MARCA (<u>nombre</u>) PK: nombre</p>
<p>CATEGORÍA_DE_ARTICULO (<u>nombre_cat,código_art</u>) PK: nombre_cat,código_art FK: nombre_cat → CATEGORÍA.nombre FK: código_art → ARTÍCULO.código</p>	<p>ALIMENTACIÓN (<u>Cód_Ali</u>) PK: Cód_Ali FK: Cód_Ali → ARTÍCULO.Código</p>
<p>BEBIDA (<u>Cód_Beb</u>) PK: Cód_Beb FK: Cód_Beb → ARTÍCULO.Código</p>	<p>DULCES_Y_SNACKS(<u>Cód_duli</u>) PK: Cód_dul FK: Cód_dul → ARTÍCULO.Código</p>
<p>POSTRES (<u>Cód_Pos</u>) PK: Cód_Pos FK: Cód_Pos → ARTÍCULO.Código</p>	<p>CAJA_SORPRESA (<u>Cód_Caja</u>) PK: Cód_Caja FK: Cód_Caja → ARTÍCULO.Código</p>

Modelo Entidad/Relación

Esquema



[Link de Lucidchart](#)

Explicación

Nuestro proyecto se basa en una tienda online de comida y bebida japonesa donde ofreceremos todo tipo de productos de este tipo ya sean bebidas, snacks, dulces, sopas etc...

Nuestra idea es realizar algo similar a la tienda Japon Shop.

En primer lugar, nos encontramos con la entidad ARTÍCULOS que se trata de una generalización exclusiva y parcial donde sus subtipos son: Alimentación, bebida, dulces y snacks, postres y caja sorpresa.

Para la entidad ARTÍCULO tenemos cinco relaciones con: Marca, Categoría, dos con Usuario y Cesta.

ARTÍCULO tiene MARCA - Se trata de una relación (1,n) 1:N (1,1) ya que un artículo como mínimo y máximo tendrá una marca y una marca como mínimo será de un producto y como máximo de muchos. Es una dependencia de existencia ya que si no existiera la marca no existiría el artículo.

ARTÍCULO bucar CATEGORÍA - Se trata de una relación (0,n) N:N (1,n) ya que un artículo podrá ser buscado en mínimo una categoría y máximo en muchas y en una categoría podrán localizarse mínimo ningún artículo y máximo muchos.

ARTÍCULO añadir CESTA - Se trata de una relación (0,n) N:N (0,n) ya que un artículo podrá añadirse mínimo en ninguna cesta y máximo en muchas y una cesta podrá tener como mínimo ningún artículo y como máximo muchos.

ARTÍCULO crear USUARIO- Se trata de una relación (0,n) 1:N (1,1) ya que un artículo podrá ser creado mínimo por un administrador y máximo por uno también y una administrador podrá crear muchos artículos o ninguno. Se trata de una dependencia de existencia porque no existirían los artículos si un administrador no los creara.

ARTÍCULO modificar USUARIO- Se trata de una relación (0,n) 1:N (1,1) ya que un artículo podrá ser modificado mínimo por un administrador y máximo por uno también y una administrador podrá modificar muchos artículos o ninguno. Se trata de una dependencia de existencia porque no existirían los artículos si un administrador no los creara.

Para la entidad CESTA tenemos tres relaciones con: Artículo, Pedido y Cliente.

CESTA añadir ARTÍCULO - Esta relación ya la hemos explicado anteriormente.

CESTA crear CLIENTE - Se trata de una relación (0,1) 1:1 (1,1) ya que una cesta es creada como mínimo por un cliente y como máximo también y un cliente podrá crear como mínimo ninguna cesta y como máximo 1 cesta.

CESTA comprar PEDIDO - Se trata de una relación (1,1) 1:1 (0,1) ya que en un pedido puedes comprar mínimo lo que hay en una cesta y máximo también y con lo que hay en una cesta puedes hacer mínimo ningún pedido y máximo uno. Existe una dependencia de existencia ya que si no existiera la cesta no se podría realizar el pedido.

Para la entidad CLIENTE tenemos cuatro relaciones con: Tarjeta, Dirección y Cesta.
CLIENTE vincula TARJETA - Se trata de una relación (0, n) N:N (0, n) ya que un cliente como mínimo puede vincular 0 tarjetas y como máximo puede vincular muchas tarjetas y una tarjeta puede ser vinculada como mínimo con 0 clientes y como máximo con muchos clientes.

CLIENTE crea CESTA - Esta relación ya la hemos explicado anteriormente.

CLIENTE asocia DIRECCIÓN - Se trata de una relación (1,1) 1:1 (1:1) ya que un cliente puede asociar mínimo una dirección y como máximo una dirección, y una dirección puede ser asociada mínimo por un cliente y máximo por un cliente.

Para la entidad PEDIDO tenemos tres relaciones con: Factura, Dirección y Cesta.

PEDIDO genera FACTURA - Se trata de una relación (1:1) 1:1 (0:1) ya que un pedido puede generar como mínimo ninguna factura y como máximo 1 factura y una factura puede ser generada mínimo por 1 pedido y máximo por 1 pedido. Tenemos una dependencia de existencia ya que podemos identificar la factura por su código sin necesidad del pedido y el pedido se puede identificar por su número de pedido sin necesidad de la factura.

PEDIDO dir-envío DIRECCIÓN - Se trata de una relación (0,n) 1:N (1:1) ya que un pedido puede tener como mínimo 1 dirección de envío y como máximo 1 dirección de envío, y una dirección puede tener de envío como mínimo 0 pedidos y como máximo muchos pedidos. Tenemos una dependencia de existencia ya que podemos identificar al pedido por su nº de pedido sin necesidad de la dirección.

DIRECCIÓN tiene a su vez una relación con FACTURA.

DIRECCIÓN dir-fact FACTURA - Se trata de una relación (0,n) 1:N (1:1) ya que una dirección puede tener como mínimo 0 facturas y como máximo muchas, y una factura puede tener como mínimo una dirección y como máximo una dirección. Tenemos una dependencia de existencia ya que podemos identificar la factura por su código sin necesidad de la dirección.

PEDIDO compra CESTA - Esta relación ya la hemos explicado anteriormente.

PROGRAMACIÓN

Servlet

LoginServlet

```
public class LoginServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try {
            response.setContentType("text/html;charset=UTF-8");
            String email = request.getParameter("usuario");
            String password = request.getParameter("password");
            // Hacemos uso de logica de negocio del bean (dto) como el método "validar()"
            Usuario usuario = new UsuarioDAO().validar(email, password);

            if(usuario != null) {
                // Si el login es OK guardamos el objeto de tipo "Usuario" en la sesión para poder recuperarlo más tarde.
                // Y tras esto redirigimos a "index.html"

                HttpSession session = request.getSession();
                session.setAttribute("usuario", usuario);
                response.sendRedirect("index.jsp");
            } else {
                // Si el login no es correcto, enviamos un atributo "error" a true a "login.jsp" para que muestre el error.
                request.setAttribute("error", true);
                request.getRequestDispatcher("login.jsp").include(request, response);
            }
        } catch (SQLException ex) {
            System.out.println("ERROR SQL");
        }
    }

}
```

Líneas y explicación

```
String email = request.getParameter("usuario");
String password = request.getParameter("password");
```

Obtenemos los parámetros de la solicitud llamados "usuario" y "password". Estos parámetros se enviaron desde el formulario de login.jsp

```
Usuario usuario = new UsuarioDAO().validar(email, password);
```

Rescatamos el usuario desde la base de datos pasando antes por el método validar el cuál devolverá null en caso de no encontrar ningún usuario que coincida

Después, en el IF-ELSE comprueba que el usuario no sea nulo, si no es nulo guarda la sesión y te redirige al index.jsp con la sesión iniciada.

LogoutServlet

```
public class LogoutServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        HttpSession session=request.getSession();
        session.invalidate();
        response.sendRedirect("index.jsp");
    }

    //HttpServlet methods. Click on the + sign on the left to edit the code.

}
```

[HttpSession session=request.getSession\(\);](#)

Obtiene la sesión actual del usuario

[session.invalidate\(\);](#)

Invalide la sesión actual.

[response.sendRedirect\("index.jsp"\);](#)

Nos redirige a la página "index.jsp".

PerfilUsuario

```
public class PerfilUsuario extends HttpServlet {  
  
    /**  
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.  
     *  
     * @param request servlet request  
     * @param response servlet response  
     * @throws ServletException if a servlet-specific error occurs  
     * @throws IOException if an I/O error occurs  
     */  
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html;charset=UTF-8");  
        try ( PrintWriter out = response.getWriter() ) {  
            HttpSession session = request.getSession(false);  
            Usuario usuarioSesion = (Usuario) session.getAttribute("usuario");  
            out.print("Codigo: " + usuarioSesion.getCodigo());  
            out.print("Nombre: " + usuarioSesion.getCodigo());  
            out.print("Fecha nacimiento: " + usuarioSesion.getCodigo());  
            out.print("Email: " + usuarioSesion.getCodigo());  
            out.print("Telefono: " + usuarioSesion.getCodigo());  
            out.print("Codigo: " + usuarioSesion.getCodigo());  
            out.print("<p><a href=\"index.html\">Volver al indice</a></p>");  
        }  
    }  
}
```

HttpSession session=request.getSession(false);

Obtiene la sesión actual del usuario. Se le pasa “false” para indicar que no hay que crear una nueva y que ya existe una sesión.

Usuario usuarioSesion = (Usuario) session.getAttribute("usuario");

Rescatamos el usuario desde el atributo de la sesión actual.

El resto de líneas, utilizando el print, mostramos los datos del usuario de la sesión

AnyadirAlCarrito

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException, SQLException {
    response.setContentType("text/html;charset=UTF-8");
    HttpSession session = request.getSession();
    PrintWriter out = response.getWriter();
    try {
        Usuario usuarioSession = (session != null && session.getAttribute("usuario") != null) ? (Usuario) session.getAttribute("usuario") : null;
        String strId = request.getParameter("id");
        String strCantidad = request.getParameter("cantidad");

        if (strId != null && strCantidad != null && session.getAttribute("usuario") != null) {
            Articulo art = new ArticuloDAO().getByCodigo(Integer.parseInt(strId));
            int cantidad = Integer.parseInt(strCantidad);
            ArrayList<LineaCesta> lineas = new ArrayList<>();
            LineaCesta linea = new LineaCesta(cantidad, art.getPvp() * cantidad, art);
            CestaDAO cesta = new CestaDAO();

            if (cesta.getByNombre(usuarioSession.getEmail()) == null) {
                lineas.add(linea);
                Cesta cesta = new Cesta(usuarioSession.getEmail(), usuarioSession, lineas);
                cesta.anyadir(cesta);
            }
        }
        try {
            cesta.anyadirLinea2(linea, cesta.getByNombre(usuarioSession.getEmail()));
        } catch (Exception e) {
        }

        response.sendRedirect("cesta.jsp");
    }
}

```

HttpSession session=request.getSession();

Obtiene la sesión actual del usuario

Usuario usuarioSession = (session != null && session.getAttribute("usuario") != null) ?
 (Usuario) session.getAttribute("usuario") : null;

Verificamos si la sesión existe y si el atributo "usuario" está presente en la sesión. Si es así, obtenemos el usuario de la sesión.

String strId = request.getParameter("id");
 String strCantidad = request.getParameter("cantidad");

Obtenemos los parámetros de la solicitud llamados "id" y "cantidad". Estos parámetros se enviaron desde el formulario de articulo.jsp

if (strId != null && strCantidad != null && session.getAttribute("usuario") != null) {

Verifica si los parámetros "id" y "cantidad" no son nulos y si el atributo "usuario" está presente en la sesión.

Articulo art = new ArticuloDAO().getByCodigo(Integer.parseInt(strId));

Recogemos el artículo indicado mediante el método getByCodigo donde le entra el "id" que es el código del producto

```
int cantidad = Integer.parseInt(strCantidad);
ArrayList<Lineacesta> lineas = new ArrayList<>();
Lineacesta linea = new Lineacesta(cantidad, art.getPvp() * cantidad, art);
CestaDAO cest = new CestaDAO();
```

Guardamos la cantidad en una variable int.

Creamos un arraylist llamado líneas

Creamos un objeto lineacesta formada por la variable cantidad, el precio del artículo multiplicado por la cantidad y el propio artículo.

Creamos un objeto CestaDAO llamado cest.

```
if (cest.getByName(usuarioSesion.getEmail()) == null) {
    lineas.add(linea);
    Cesta cesta = new Cesta(usuarioSesion.getEmail(), usuarioSesion, lineas);
    cest.anyadir(cesta);

}
```

Comprobamos si el cliente ya tiene una cesta creada mediante el método getByName() el cuál localiza una cesta pasándole el nombre de dicha cesta, en nuestro caso, el nombre de la cesta será el email del usuario ya que no se puede repetir.

En caso de que devuelva null, lo que quiere decir que ese cliente no tiene ninguna cesta creada todavía, entrará al if y realizará lo siguiente:

- Añadirá la lineacesta a la base de datos mediante el método add()
- Creará una nueva cesta utilizando de nombre el email del usuario y como código de cliente el código del usuario de la sesión
- Añadirá la cesta creada en la línea anterior a la base de datos mediante el método anyadir().

```
cest.anyadirLinea2(linea, cest.getByName(usuarioSesion.getEmail()));
```

Si no devolviera null, lo que quiere decir que ese cliente ya tiene una cesta creada, simplemente añadirá la linea a la base de datos mediante el método anyadirLinea2 donde le entra la propia linea además de la cesta la cual será rescatada de la base de datos utilizando el método getByName() pasándole el nombre del cliente el cuál será el email del usuario de la sesión.

```
response.sendRedirect("cesta.jsp");
```

Después nos redirigirá a cesta.jsp.

GuitarDelCarrito

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException, SQLException {
    response.setContentType("text/html;charset=UTF-8");
    HttpSession session = request.getSession();
    try ( PrintWriter out = response.getWriter() ) {
        Usuario usuarioSesion = (session != null && session.getAttribute("usuario") != null) ? (Usuario) session.getAttribute("usuario") : null;
        Articulo a = new ArticuloDAO().getByCodigo(Integer.parseInt(request.getParameter("id")));
        CestaDAO c = new CestaDAO();
        c.eliminarLinea(a.getCodigo(), c.getByNombre(usuarioSesion.getEmail()));
        response.sendRedirect("cesta.jsp");
    }
}
```

HttpSession session=request.getSession();

Obtiene la sesión actual del usuario

Usuario usuarioSesion = (session != null && session.getAttribute("usuario") != null) ?
(Usuario) session.getAttribute("usuario") : null;

Verificamos si la sesión existe y si el atributo "usuario" está presente en la sesión. Si es así, obtenemos el usuario de la sesión.

```
Articulo a = new ArticuloDAO().getByCodigo(Integer.parseInt(request.getParameter("id")));
CestaDAO c = new CestaDAO();
c.eliminarLinea(a.getCodigo(), c.getByNombre(usuarioSesion.getEmail()));

response.sendRedirect("cesta.jsp");
```

Recogemos el artículo indicado mediante el método getByCodigo donde le entra el "id" que le llega como parámetro cuando se acciona el servlet.

Creamos una nueva cestaDAO llamada c.

Eliminamos la línea de la base de datos mediante el método eliminarLinea al cuál le entra el código del artículo creado anteriormente y la cesta que se localizará mediante el getByNombre() pasandole el nombre de la cesta que será el email del usuario de la sesión.

ProcesarPago

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException, SQLException {
    response.setContentType("text/html;charset=UTF-8");
    HttpSession session = request.getSession();

    try ( PrintWriter out = response.getWriter() ) {
        Usuario usuarioSession = (session != null && session.getAttribute("usuario") != null) ? (Usuario) session.getAttribute("usuario") : null;
        CestaDAO cesta = new CestaDAO();
        DireccionDAO direccion = new DireccionDAO();

        Cesta cest = cesta.getByNombre(usuarioSesion.getEmail());

        Direccion dir = direccion.getDireccionDe(usuarioSesion);

        PedidoDAO pedido = new PedidoDAO();
        int siguienteCodigo = pedido.siguienteCodigo();
        Pedido ped = new Pedido(siguienteCodigo, LocalDateTime.now(), cest, dir);

        pedido.anyadir(ped);

        FacturaDAO factura = new FacturaDAO();
        int sigCod = factura.siguienteCodigo();
        Factura fac = new Factura(sigCod, LocalDateTime.now(), ped, dir);
        factura.anyadir(fac);
        Cesta ces2 = new Cesta(String.valueOf(siguienteCodigo), usuarioSesion, cest.getLineas());
        cesta.anyadir(ces2);

        ArrayList<Lineacesta> lineas = cesta.getLineas(usuarioSesion.getEmail(), usuarioSesion.getCodigo());
        for (Lineacesta linea : lineas) {
            cesta.anyadirLinea2(linea, ces2);
            cesta.eliminarLinea(linea.getArticulo().getCodigo(), cest);
        }

        response.sendRedirect("../pedidoRealizado.jsp");
    }
    catch (Exception ex) {
        ex.getMessage();
    }
}
```

CestaDAO cesta = new CestaDAO();
DireccionDAO direccion = new DireccionDAO();

Creamos un objeto cestaDAO y otro DireccionDAO

Cesta cest = cesta.getByNombre(usuarioSesion.getEmail());
Direccion dir = direccion.getDireccionDe(usuarioSesion);

Creamos una cesta mediante el método getByNombre() el cuál localiza una cesta en la base de datos pasándole el nombre de dicha cesta, en nuestro caso, el nombre de la cesta será el email del usuario ya que no se puede repetir.

Creamos una dirección mediante el código getDirección el cuál localiza una dirección en la base de datos pasándole el usuario de la sesión, el método buscará la dirección asociada a ese usuario y la guardará en el objeto dirección llamado dir.

```
PedidoDAO pedido = new PedidoDAO();
int siguienteCodigo = pedido.siguienteCodigo();
Pedido ped = new Pedido(siguienteCodigo, LocalDateTime.now(), cest, dir);
pedido.anyadir(ped);
```

Creamos un objeto PedidoDAO.

Creamos una variable de tipo int para que en ella guarde el código del último pedido más 1. Creamos un nuevo pedido pasándole como código la variable siguiente pedido, como fecha la fecha actual, como cesta la cesta “cest” creada anteriormente y como dirección la dirección “dir” también creada anteriormente.

Añadimos el pedido creado a la base de datos mediante el método anyadir()

```
FacturaDAO factura = new FacturaDAO();
int sigCod = factura.siguienteCodigo();
Factura fac = new Factura(sigCod, LocalDateTime.now(), ped, dir);
factura.anyadir(fac);
```

Creamos un objeto FacturaDAO.

Creamos una variable de tipo int para que en ella guarde el código del último pedido más 1. Creamos una nueva factura pasándole como código la variable sigCod, como fecha la fecha actual, como pedido el pedido “ped” creado anteriormente y como dirección la dirección “dir” también creada anteriormente.

Añadimos la factura creada a la base de datos mediante el método anyadir()

```
Cesta ces2 = new Cesta(String.valueOf(siguienteCodigo) , usuarioSesion, cest.getLineas());
cesta.anyadir(ces2);
```

Creamos una cesta copia que nos servirá luego más tarde porque a la hora de que el cliente procese el pago, todas las líneas de la cesta original se borrarán para que cuando haga otro pedido no aparezcan por lo que esta cesta copia nos permite mantener las líneas de esa cesta y utilizar esos datos cuando nos sea necesario más adelante.

A la cesta copia le pasamos la variable siguienteCodigo que coincidirá siempre con el código pedido por lo que esta cesta la podremos localizar siempre con el código de su pedido, el usuario de la sesión y un arraylist de lineas que sacaremos de la cesta original mediante el método getLineas().

Luego añadimos esa cesta a la base de datos mediante el método anyadir.

```
ArrayList<Lineacesta> lineas = cesta.getLineas(usuarioSesion.getEmail(),
usuarioSesion.getCodigo());
for (Lineacesta linea : lineas ) {
    cesta.anyadirLinea2(linea, ces2);
    cesta.eliminarLinea(linea.getArticulo().getCodigo(), cest);
}
response.sendRedirect("./pedidoRealizado.jsp");
```

Estas líneas se encargan de lo mencionado anteriormente, guardamos un arraylist de las líneas de la cesta del usuario y mediante el bucle for añadimos cada línea a la cesta copia y la eliminamos de la cesta del cliente para que quede vacía para próximas compras.

Después nos redirigirá a pedidoRealizado.jsp.

JSP

index

```
<nav>
    <section class="menu">
        <ul class="nav-list">
            <li class="navegar"><a href="../productos.jsp">Productos</a></li>
            <li class="navegar"><a href="../concurso.jsp">Concurso</a></li>
            <li class="navegar"><a href="../contacto.jsp">Contacto</a></li>
            <li class="navegar"><a href="../sobrenosotros.jsp">Sobre Nosotros</a></li>
            <li><p><input type="search" name="buscador" placeholder="Buscar en San-Je Shop" class="buscador"/></p></li>
        <%
            Usuario usuarioSesion = (session != null && session.getAttribute("usuario") != null) ? (Usuario) session.getAttribute("usuario") : null;
            if (usuarioSesion != null) {
        %>
            <li class="inicio-sesion"><a href="../perfil.jsp"><%= usuarioSesion.getNombreCompleto() %></a></li>
        <%
                if (usuarioSesion.getTipo() == TipoUsuario.CLIENTE) {
        %>
                    <a href="../cesta.jsp">
                        <figure class="cesta">
                            
                        </figure>
                    </a>
                    <%
                }
                else{
        %>
            <li class="inicio-sesion"><a href="../login.jsp">Entrar / Registrarse</a></li>
        <%
        }
        <%
    </ul>
</section>
</nav>
```

Usuario usuarioSesion = (session != null && session.getAttribute("usuario") != null) ?
 (Usuario) session.getAttribute("usuario") : null;

Se crea una variable llamada usuarioSesion, la cual como valor saca una comprobación: saca si la sesión es distinta a null y el atributo usuario de esa misma sesión es distinto a null entonces se guarda en la variable el valor del atributo usuario de la sesión, si no se guarda el valor null.

if (usuarioSesion != null)

Se hace una comprobación para saber si la sesión tiene un usuario o si es nula, utilizando la variable creada en el paso anterior. En caso de ser distinta a null, en la barra de navegación aparecerá el nombre del usuario en sesión (<%= usuarioSesion.getNombreCompleto() %>)

else

En caso de que el usuarioSesion sea nulo (no haya ninguna sesión activa) en lugar de mostrarse en el menú de navegación el nombre del usuario se mostrará el botón Entrar/Registrarse.

if (usuarioSesion.getTipo() == TipoUsuario.CLIENTE)

Tenemos una segunda comprobación una vez que la sesión ya está activa, que lo que hace es comprobar si el usuario que está en la sesión, es de tipo cliente, comprobando el enum. En caso de ser true, al lado del nombre del usuario en la barra de navegación aparecerá también el icono de la cesta para poder realizar compras.

articulo

```
<%
    Articulo articulo;
    int codigo = Integer.parseInt(request.getParameter("id"));
    articulo = new ArticuloDAO().getByCodigo(codigo);
%>

<body>

<div id="contenedor">
    <div id="central">
        <div id="login">
            <div class="titulo">
                <%= articulo.getNombre() %>
            </div>
            <div class="img">
                
            </div>
            <div class="foto-perfil">
                
            </div>
            <p><%= articulo.getPvp() %></p>
            <p class="desc"><%= articulo.getDescripcion() %></p>
        <%
            Usuario usuarioSession = (session != null && session.getAttribute("usuario") != null) ? (Usuario) session.getAttribute("usuario") : null;
            if (usuarioSession != null) {
        %>
            <form name="anyadircarrrito" method="post" action="AnyadirAlCarrito">
                <input type="hidden" name="id" value="<%= articulo.getCodigo() %>" />
                <p>Cantidad: <input type="number" id="cantidad" name="cantidad" min="1" max="100" value="1" class="cantidad"></p>
                <div class="pie-form">
                    <input class="añadir" type="submit" value="Añadir"/>
                </div>
            </form>
        <%
            }
            else{
        %>
            <p><strong>Necesitas iniciar sesión para añadirlo al carrito.</strong></p>
            <div class="pie-form">
                <a href=".login.jsp">Iniciar Sesión</a>
            </div>
        <%
            }
        <%
    </div>
</body>
```

Articulo articulo;

Crea una variable llamada articulo, sin ningún valor inicial.

`int codigo = Integer.parseInt(request.getParameter("id"));`

Crea una variable llamada codigo, que saca el valor pidiendo de la sesión el atributo id, y con el metodo parseInt convierte dicho id que viene como texto en un número.

`articulo = new ArticuloDAO().getByCodigo(codigo);`

Le da un valor a la variable articulo creando un nuevo ArticuloDAO con el código rescatado o pedido de la sesión con el paso anterior.

`articulo.getNombre()`

Saca el nombre del artículo con el método getNombre realizado en el dto.

`String foto = articulo.getFoto();`

`if (foto == null)`

Guarda en una variable de tipo texto el valor que obtiene sacando la foto del articulo con el método getFoto realizado en el dto. Posteriormente, hace una comprobación de si el valor de foto es null. En caso de ser null, muestra una foto genérica del artículo.

cesta

```

<main>
    <section class="titulo-main">
        <h2 class="tit">Cesta de <%= usuarioSesion.getNombreCompleto()%></h2>
        <figure class="cesta">
            
        </figure>
    </section>
    <%
        CestaDAO cesta = new CestaDAO();
        ArrayList<Lineacesta> lineas = cesta.getLineas(usuarioSesion.getEmail(), usuarioSesion.getCodigo());
    %>
    <section class="product-detail">

        <div class="my-order-content">
            <% double acumulado = 0;%>
            <% for (Lineacesta linea : lineas) {%
                <div class="shopping-cart">
                    <% String foto = linea.getArticulo().getFoto(); %>
                    <%
                        if (foto == null) {
                    %>
                    <figure>
                        
                    </figure>
                    <%
                } else {
                    <figure>
                        
                    </figure>
                    <%
                }
            %>
            <p class="nom"><%= linea.getArticulo().getNombre()%></p>
            <p class="prec"><%= linea.getArticulo().getPvp()%>€</p>
            <% acumulado += linea.getTotalLinea();%>
            <p class="cant"><%= linea.getCantidad()%> Uds</p>
            <a href="/QuitarDelCarrito?id=<%= linea.getArticulo().getCodigo()%>" class="eliminar">Eliminar</a>
        </div>
        <% }%>
        <% if(!lineas.isEmpty()) {%
            <div>
                <p class="total"><strong>Total sin IVA: </strong><%= Math.round((acumulado) * 100)/100,%>€ </p>
                <p class="total"><strong>TOTAL: </strong><%= Math.round((acumulado + acumulado * 0.21) * 100)/100,%>€ </p>
            </div>
            <a class="primary-button" href="/ProcesarPago">Procesar el pago</a>
            <%
            }
            else {
            %>
            <p class="vacio">Su cesta está vacía, añada productos para realizar un pedido.</p>
            <%
            }
        %>
        </div>
    </div>
</section>
</main>

```

CestaDAO cesta = new CestaDAO();

Crea una variable de tipo CestaDAO llamada cesta, con el valor de una nueva cestaDAO.

ArrayList<Lineacesta> lineas = cesta.getLineas(usuarioSesion.getEmail(),
usuarioSesion.getCodigo());

Crea un ArrayList de Lineacesta llamada líneas, y como valor saca las líneas de la cesta pasandole como valores el usuario de la sesión y su código.

double acumulado = 0;

Crea una variable llamada acumulado con el valor 0.

for (Lineacesta linea : lineas)

Crea un bucle de lineacesta que por cada línea que encuentre del arraylist líneas la muestra en la cesta del cliente.

if(!lineas.isEmpty())

Comprueba que el array de líneas no esté vacío.

editar-articulo

```
<%  
    Articulo articulo;  
    int codigo = Integer.parseInt(request.getParameter("id"));  
    articulo = new ArticuloDAO().getByCodigo(codigo);  
%>  
  
<body>  
  
    <div id="contenedor">  
        <div id="central">  
            <div id="login">  
                <div class="titulo">  
                    <%= articulo.getNombre() %>  
                </div>  
                <% String foto = articulo.getFoto(); %>  
                <%  
                    if (foto == null) {  
                %>  
                    <div class="img">  
                          
                    </div>  
                <%  
                    } else {  
                %>  
                      
                <%  
                }  
                <%>  
                <p>Nuevo precio: <input type="number" id="precio" name="precio"  
                    min="1" max="100" class="cantidad" step="0.01"></p>  
                <textarea rows="6" cols="50" name="descripcion" id="descripcion" placeholder="Nueva descripción"></t  
                <p>Cantidad: <input type="number" id="cantidad" name="cantidad"  
                    min="1" max="100" class="cantidad"></p>  
                <div class="pie-form">  
                    <input class="boton" type="submit" value="Confirmar edición"/>  
                </div>  
            </div>  
            <div class="inferior">  
                <a href=".//productos.jsp">Volver</a>  
            </div>  
        </div>  
    </div>
```

Articulo articulo;

Crea una variable llamada articulo, sin ningún valor inicial.

`int codigo = Integer.parseInt(request.getParameter("id"));`

Crea una variable llamada codigo, que saca el valor pidiendo de la sesión el atributo id, y con el método parseInt convierte dicho id que viene como texto en un número.

`articulo = new ArticuloDAO().getByCodigo(codigo);`

Le da un valor a la variable artículo creando un nuevo ArticuloDAO con el código rescatado o pedido de la sesión con el paso anterior.

login

```
<div id="contenedor">
    <div id="central">
        <div id="login">
            <div class="titulo">
                Bienvenido a San-Je Shop
            </div>
            <form id="loginform" action="LoginServlet" method="post">
                <input type="text" name="usuario" placeholder="Usuario" required>

                <input type="password" placeholder="Contraseña" name="password" required>
                <input class="boton" type="submit" value="Iniciar Sesión"/>
                <%
                    if (request.getAttribute("error") != null && (boolean) request.getAttribute("error")) {
                        out.println("<p>(*) Nombre de correo o contraseña inválidos. Vuelve a intentarlo</p>");
                    }
                %>
            </form>
            <div class="pie-form">
                <a href="#">¿Perdiste tu contraseña?</a>
                <a href=".//registrarse.html">¿No tienes Cuenta? Regístrate</a>
            </div>
        </div>
        <div class="inferior">
            <a href=".//index.jsp">Volver</a>
        </div>
    </div>
</div>
```

```
if (request.getAttribute("error") != null && (boolean) request.getAttribute("error")) {
    out.println("<p>(*) Nombre de correo o contraseña inválidos. Vuelve a intentarlo</p>");
}
```

Recoge el atributo error y comprueba que sea distinto a null, además lo convierte en boolean y si es true entonces imprime en el menú de login una frase que dice: Nombre de correo o contraseña inválidos. Vuelve a intentarlo

pedidos-admin

```
<%
 PedidoDAO pedido = new PedidoDAO();
 ArrayList<Pedido> pedidos = pedido.getAll();
 %>
<section class="product-detail">

    <div class="my-order-content">
        <% if (!pedidos.isEmpty()) { %>
            <% for (Pedido pedi : pedidos) {
                ArrayList<Lineacesta> l = new CestaDAO().getLineas(String.valueOf(pedi.getCodigo()), pedi.getDireccion().getUsuario().getCodigo());
                double acu = 0;
                for (Lineacesta lin : l) {
                    acu += lin.getTotalLinea();
                }
            %>
        </div>
        <div class="shopping-cart">
            <div class="pedido-superior">
                <p class="cod">Codigo: <%=pedi.getCodigo()%></p>
                <p class="nom">Cliente: <%=pedi.getCesta().getUsuario().getNombreCompleto()%></p>
                <p class="fecha">Fecha: <%=pedi.getFecha().format(DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss"))%></p>
                <p class="precio">Precio: <=%Math.round((acu + acu * 0.2) * 100)/100.%></p>
            </div>
            <div class="pedido-inferior">
                <p class="dir"><%=pedi.getDireccion().getTipo() + " " + pedi.getDireccion().getDireccion() + ", " + pedi.getDireccion().getCp() + " " + pedi.getDireccion().getProvincia()%></p>
            </div>
        </div>
        <% }%>
        <% } else { %>
            <p class="vacio">Todavía no se ha realizado ningún pedido</p>
        <% } %>
    </div>
<% }%>
```

PedidoDAO pedido = new PedidoDAO();

Crea una nueva variable de tipo PedidoDAO llamada pedido y como valor un nuevo pedidoDAO.

ArrayList<Pedido> pedidos = pedido.getAll();

Crea un arraylist de Pedido llamada pedidos y como valor le pasa el método getAll realizado en PedidoDAO (Saca todos los pedidos de la base de datos).

if(!pedidos.isEmpty())

Comprueba que el arraylist no esté vacío.

for (Pedido pedi : pedidos)

Crea un bucle que pase por todos los pedi de el arraylist pedidos.

ArrayList<Lineacesta> l = new CestaDAO().getLineas(String.valueOf(pedi.getCodigo()), pedi.getDireccion().getUsuario().getCodigo());

Crea un nuevo ArrayList llamado l, en el valor hace una nueva CestaDAO sacandole las líneas con el getLineas que recibe el valor de el código del pedido en String, y también saca la direccion al pedi, saca al usuario de la dirección y saca el código de la dirección.

double acu = 0;

Crea una nueva variable llamada acu con el valor inicial a 0.

for (Lineacesta lin : l) {
 acu += lin.getTotalLinea();
 }

Hacer un bucle en el que por cada línea que encuentre, saca el precio total de dicha línea y lo suma a la variable acu.

pedidos-cliente

```
<%>
 PedidoDAO pedido = new PedidoDAO();
 ArrayList<Pedido> pedidos = pedido.getByCliente(usuarioSesion.getCodigo());
```



```
<%>
<section class="product-detail">
```

```
    <div class="my-order-content">
        <% if(!pedidos.isEmpty()) { %>
            <% for (Pedido pedi : pedidos) {
                ArrayList<Lineacesta> l = new CestaDAO().getLineas(String.valueOf(pedi.getCodigo()) , usuarioSesion.getCodigo());
                double acu = 0;
                for (Lineacesta lin : l) {
                    acu += lin.getTotalLinea();
                }
            %>
            <div class="shopping-cart">
                <div class="pedido-superior">
                    <p class="cod">Código: <%=pedi.getCodigo() %></p>
                    <p class="fecha">Fecha: <%=pedi.getFecha().format(DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss")) %></p>
                    <p class="prec">Precio: <%=Math.round((acu + acu * 0.21) * 100)/100.%>€</p>
                </div>
                <div class="pedido-inferior">
                    <a href=".//GenerarPDF?id=<%=pedi.getCodigo() %>" class="generar">Generar PDF</a>
                    <a href=".//GenerarXML?id=<%=pedi.getCodigo() %>" class="generar">Generar XML</a>
                </div>
            </div>
            <% }%>
            <% }else { %>
                <p class="vacio">Todavia no ha realizado ningn pedido</p>
            <% } %>
        </div>
    <% }%>
```

```
</div>
```

PedidoDAO pedido = new PedidoDAO();

Crea una nueva variable de tipo PedidoDAO llamada pedido, con el valor de un nuevo PedidoDAO.

ArrayList<Pedido> pedidos = pedido.getByCliente(usuarioSesion.getCodigo());

Crea un nuevo ArrayList de pedido llamado pedidos y como valor coge el método getByCliente del pedido y le pasa como parámetro el resultado del método getCodigo del usuario que hay en sesión.

ArrayList<Lineacesta> l = new CestaDAO().getLineas(String.valueOf(pedi.getCodigo()) , usuarioSesion.getCodigo());

Crea un nuevo ArrayList llamado l, en el valor hace una nueva CestaDAO sacandole las líneas con el getLineas que recibe el valor de el código del pedido en String, y también saca el código del usuario que hay en sesión.

productos

```

<%
    ArrayList<Articulo> articulos;
    articulos = new ArticuloDAO().getByTipo(request.getParameter("categoria"));
%>
<nav class="submen">
    <section>
        <ul class="enlaces-submen">
            <li class="submen boton"><a href=". /productos.jsp">Todo</a></li>
            <li class="submen boton"><a href=". /productos.jsp?categoria=Alimentacion">Alimentacion</a></li>
            <li class="submen boton"><a href=". /productos.jsp?categoria=Bebida">Bebida</a></li>
            <li class="submen boton"><a href=". /productos.jsp?categoria=Dulces_Snacks">Dulces y Snacks</a></li>
            <li class="submen boton"><a href=". /productos.jsp?categoria=Postres">Postres</a></li>
            <li class="submen boton"><a href=". /productos.jsp?categoria=Caja_Sorpresa">Caja Sorpresa</a></li>
        </ul>
    </section>
</nav>
<section class="productos">
    <% for (Articulo a : articulos) {%
    <article class="producto">
        <div class="card">
            <div class="card-border-top">
            </div>
            <% String foto = a.getFoto(); %>
            <%
                if (foto == null) {
            %>
                <div class="img">
                    
                </div>
            <%
                } else {
            %>
                <div class="img">
                    <img src=<%= foto%> alt="Foto del producto">
                </div>
            <%
                }
            %>
            <span><%= a.getNombre()%></span>
            <p class="precio"><%= a.getPvp()%>€ Sin IVA</p>
            <%
                if (usuarioSesion != null) {
            %>
                <section class="pie-producto">
                    <%
                        if (usuarioSesion.getTipo() == TipoUsuario.ADMINISTRADOR) {
                    %>
                        <a href=". /editar-articulo.jsp?id=<%=a.getCodigo()%>"><button>Editar</button></a>
                    <%
                        } else {
                    %>
                        <a href=". /articulo.jsp?id=<%=a.getCodigo()%>""><button>Añadir</button></a>
                    <%
                        }
                    %>
                </section>
            <%
                } else {
            %>
                <a href=". /articulo.jsp?id=<%=a.getCodigo()%>""><button>Ver más</button></a>
            <%
                }
            %>
        </div>
    </article>
    <% }%>
</section>

```

ArrayList<Articulo> articulos;

Crea un nuevo ArrayList de articulo llamado articulos.

```
articulos = new ArticuloDAO().getByTipo(request.getParameter("categoria"));
```

Le asigna un valor a artículos, creando un nuevo ArticuloDAO utilizando el metodo getByTipo para sacarle el tipo y le pasa el valor que recibe del parámetro categoría.

```
if (usuarioSesion.getTipo() == TipoUsuario.ADMINISTRADOR)
```

Comprueba que el usuario que hay en la sesión sea de tipo administrador.

Lenguaje de marcas

Mockup

Menu navegación



```
<li class="navegar"><a href=".productos.jsp">Productos</a></li>
<li class="navegar"><a href=".concurso.jsp">Concurso</a></li>
<li class="navegar"><a href=".contacto.jsp">Contacto</a></li>
<li class="navegar"><a href=".sobrenosotros.jsp">Sobre Nosotros</a></li>
```

Estos son los botones que aparecen en el menú de navegación, tenemos Productos, Concurso, Contacto, SobreNosotros. Que conducen a productos.jsp, concurso.jsp, contacto.jsp, sobreNosotros.jsp respectivamente.

Footer



En el footer tenemos 4 logos de redes sociales:

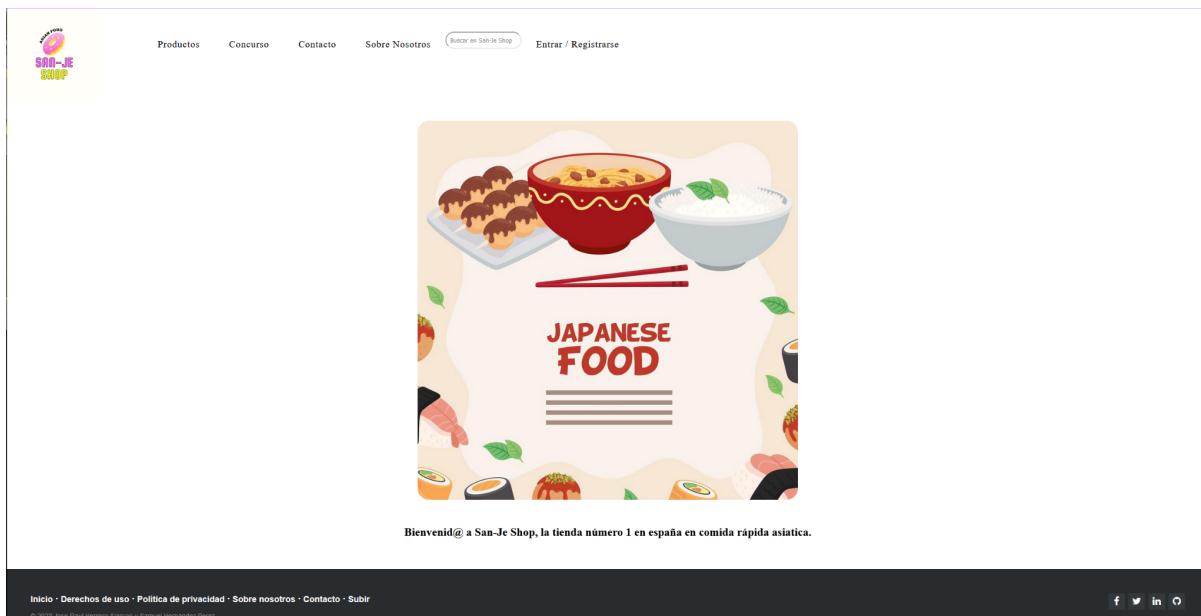
```
<a href="#"><i class="fa fa-facebook"></i></a>
<a href="#"><i class="fa fa-twitter"></i></a>
<a href="#"><i class="fa fa-linkedin"></i></a>
<a href="https://github.com/JoseRaulYSamuel"><i class="fa fa-github"></i></a>
```

En este caso, solo el icono de "fa fa-github" tiene su funcionalidad, que es redirigir a nuestro github para poder visualizar todos nuestros proyectos.

```
<a class="link-1" href=".index.jsp">Inicio</a>
<a href=".privacidad.html#derechos">Derechos de uso</a>
<a href=".privacidad.html#politica">Política de privacidad</a>
<a href=".sobrenosotros.jsp">Sobre nosotros</a>
<a href=".contacto.jsp">Contacto</a>
<a href="#">Subir</a>
```

Este menú de navegación y este footer van a ser el mismo para todas las demás páginas.

index



The screenshot shows the homepage of San-Je Shop. At the top, there is a navigation bar with links for 'Productos', 'Concurso', 'Contacto', 'Sobre Nosotros', a search bar labeled 'Buscar en San-Je Shop', and 'Entrar / Registrarse'. Below the navigation is a large, central banner featuring a variety of Asian dishes like dumplings, rice, and sushi, with the text 'JAPANESE FOOD' prominently displayed in the center. Below the banner, a welcome message reads: 'Bienvenid@ a San-Je Shop, la tienda número 1 en españa en comida rápida asiatica.' At the bottom of the page is a dark footer bar containing links for 'Inicio', 'Derechos de uso', 'Política de privacidad', 'Sobre nosotros', 'Contacto', 'Subir', and social media icons for Facebook, Twitter, LinkedIn, and YouTube.

En el index tenemos una imagen principal de bienvenida en el medio con una frase en el medio dando la bienvenida, es un index muy limpio y claro sin mucha carga.

A todo esto se le añade el menú de navegación explicado anteriormente y el footer también explicado anteriormente.



Derechos de uso y política de privacidad

Política de Privacidad

San-Je Shop te informa sobre su Política de Privacidad respecto del tratamiento y protección de los datos de carácter personal de los usuarios y clientes que puedan ser recabados por la navegación o contratación de servicios a través del sitio Web www.san-je.es.

En este sentido, San-Je Shop garantiza el cumplimiento de la normativa vigente en materia de protección de datos personales, reflejada en la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y de Garantía de Derechos Digitales (LOPD GDD). Cumple también con el Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo de

27 de abril de 2016, sobre el tratamiento de datos personales y la protección de la privacidad de las personas físicas en el contexto de la libre circulación de estos datos (Reglamento General de Protección de Datos).

El uso de este Web implica la aceptación de esta Política de Privacidad así como las condiciones incluidas en el Aviso Legal.

Identidad del responsable

• San-Je Shop: San-Je Shop

• Tfno.: +34 963 20 00 00

• Domicilio: Polígono de campo alto, 17

• Correo electrónico: contacto@san-je.es

• Sitio web: www.san-je.es

Principios aplicados en el tratamiento de datos

En el tratamiento de los datos personales, San-Je Shop aplica los siguientes principios que se ajustan a las exigencias del nuevo reglamento europeo de protección de datos:

Principio de transparencia:

San-Je Shop siempre requerirá el consentimiento para el tratamiento de tus datos personales que puede ser para uno o varios fines específicos sobre los que te informará previamente con absoluta transparencia.

Principio de minimización:

San-Je Shop solo recopilará la información necesaria para el fin o los fines que los solicita.

Principio de limitación del plazo de conservación:

Tus datos se manejaremos durante el tiempo estrictamente necesario para el fin o los fines del tratamiento.

San-Je Shop garantizará la permanencia de los datos correspondiente según la finalidad. En el caso de inscripciones, San-Je Shop revisará periódicamente las listas y eliminará aquellos registros inactivos durante un tiempo considerable.

Principio de integridad y confidencialidad:

Tus datos serán tratados de tal manera que sea separada, confidencialidad e integridad esté garantizada. Debes saber que San-Je Shop toma las precauciones necesarias para evitar el acceso no autorizado o uso indebido de los datos de sus usuarios por parte de terceros.

Principio de datos personales:

Podrás acceder, rectificar o suprimir tus datos personales. Los casos en los que te informaremos tus datos personales son los siguientes:

Al contactar a través de los formularios de contacto o enviar un correo electrónico. Al realizar un comentario en un artículo o página. Al inscribirte en un formulario de suscripción o un boletín que el San-Je Shop gestiona con MailChimp. Al inscribirte en un formulario de suscripción o un boletín que el San-Je Shop gestiona con MailRelay. Al inscribirte en un formulario de suscripción o un boletín que el San-Je Shop gestiona con Semrush.

Tu derecho a la información:

San-Je Shop te informa sobre tus datos personales teniendo derecho a:

• Solucionar el acceso a los datos almacenados.

• Solucionar la imprecisión de los datos.

• Solucionar la omisión de los datos.

• Solucionar la exceso de los datos.

El ejercicio de estos derechos es personal y por tanto debe ser ejercido directamente por el interesado, solucionando directamente a San-Je Shop, lo que significa que cualquier cliente, empleado o colaborador que haya facilitado sus datos en algún momento puede dirigirse a San-Je Shop y pedir información sobre los datos que tiene almacenados y cómo los ha tratado, solicitar la rectificación de los mismos, solicitar la limitación o la supresión de los mismos y en su caso dirigirse al Juzgado de Instrucción de la localidad en la que reside o al Juzgado de lo Social de la localidad en la que trabaja.

Para ejercer tu derecho a la información, debes dirigirte a San-Je Shop, en la dirección que figura en el apartado de contacto de este documento.

Tus derechos a la medida judicial efectiva y a presentar una reclamación ante la autoridad de control, en este caso, la Agencia Española de Protección de Datos, si consideras que el tratamiento de datos personales que te conciernen infringe el Reglamento.

Finalidad del tratamiento de datos personales:

Como usuario de este sitio Web, te informamos que se recopilan y se procesan los datos que te proporcionas para el funcionamiento de nuestro sitio Web. Esta información puede incluir datos de carácter personal como pueden ser tu dirección IP, nombre y apellido, dirección física, dirección de correo electrónico, número de teléfono, y otra información. Al facilitarte esta información, das tu consentimiento para que tu información sea recopilada, utilizada, gestionada y almacenada por superávit, así como se describe en el Aviso Legal y en la presente Política de Privacidad.

Los datos personales y la finalidad del tratamiento por parte de San-Je Shop:

San-Je Shop solicita datos personales entre los que puedes tener: Nombre y apellido, dirección de correo electrónico, número de teléfono y dirección de tu sitio Web con la finalidad de responder a tus consultas.

Por ejemplo, San-Je Shop utiliza estos datos para dar respuesta a tus mensajes, llamadas, correas, comentarios o inquietudes que puedas tener relativas a la información incluida en el sitio Web, los servicios que se prestan a través del sitio Web, el tratamiento de tus datos personales, cuestiones referentes a los textos legales incluidos en el sitio Web, así como cualquier otra información que nos pases.

Formularios de suscripción a contenidos:

Los datos que facilitas en los formularios de suscripción a contenidos, Nombre y apellido, dirección de correo electrónico, número de teléfono y dirección de tu sitio web para prestarles la lista de suscripciones, enviar boletines, promociones y ofertas especiales.

Los datos personales y la finalidad del tratamiento por parte de San-Je Shop:

Los datos que facilitas en los formularios de suscripción a contenidos, Nombre y apellido, dirección de correo electrónico, número de teléfono y dirección de tu sitio web para prestarles la lista de suscripciones, enviar boletines, promociones y ofertas especiales.

Existencia otras finalidades para las que San-Je Shop trata tus datos personales:

Para analizar la navegación. San-Je Shop recoge otros datos no identificativos que se obtienen mediante el uso de cookies para dar respuesta a tu solicitud cuando navegas por el sitio Web cuyas características y finalidad están detalladas en la Política de Cookies.

Para mejorar y optimizar los servicios que ofrece este sitio Web:

Para analizar la navegación. San-Je Shop recoge otros datos no identificativos que se obtienen mediante el uso de cookies para dar respuesta a tu solicitud cuando navegas por el sitio Web cuyas características y finalidad están detalladas en la Política de Cookies.

Para analizar la navegación. San-Je Shop recoge otros datos no identificativos que se obtienen mediante el uso de cookies para dar respuesta a tu solicitud cuando navegas por el sitio Web cuyas características y finalidad están detalladas en la Política de Cookies.

San-Je Shop tratará tus datos personales con la finalidad de administrar correctamente su presencia en la red social, informarte de sus actividades, productos o servicios, así como para cualquier otra finalidad que las normativas de las redes sociales permitan.

San-Je Shop te informa que se recopilan y se procesan datos personales para la administración de las redes sociales para enviar publicidad de manera individual.

Suscríbete a las listas de datos personales:

Para proteger tus datos personales, San-Je Shop toma todas las precauciones razonables y sigue las mejores prácticas de la industria para evitar su pérdida, mal uso, acceso indebido, divulgación, alteración o destrucción de los mismos.

Para analizar la navegación. San-Je Shop recoge otros datos no identificativos que se obtienen mediante el uso de cookies para dar respuesta a tu solicitud cuando navegas por el sitio Web cuyas características y finalidad están detalladas en la Política de Cookies.

Contenido de otros sitios Web:

Para proteger tus datos personales, San-Je Shop toma todas las precauciones razonables y sigue las mejores prácticas de la industria para evitar su pérdida, mal uso, acceso indebido, divulgación, alteración o destrucción de los mismos.

Contenido de otros sitios Web:

Para proteger tus datos personales, San-Je Shop toma todas las precauciones razonables y sigue las mejores prácticas de la industria para evitar su pérdida, mal uso, acceso indebido, divulgación, alteración o destrucción de los mismos.

Política de Cookies:

Para proteger tus datos personales, San-Je Shop toma todas las precauciones razonables y sigue las mejores prácticas de la industria para evitar su pérdida, mal uso, acceso indebido, divulgación, alteración o destrucción de los mismos.

Legitimación para el tratamiento de datos:

La base legal para el tratamiento de tus datos es el consentimiento.

Política de tratamiento de datos personales:

Para proteger tus datos personales, San-Je Shop toma todas las precauciones razonables y sigue las mejores prácticas de la industria para evitar su pérdida, mal uso, acceso indebido, divulgación, alteración o destrucción de los mismos.

Categorías de datos personales:

Las categorías de datos personales que trata San-Je Shop son:

Datos personales:

Los datos personales que proporcionas a San-Je Shop se conservarán hasta que solicites su supresión.

Destinatarios de datos personales:

Google Analytics: Google Analytics es una aplicación web prestada por Google, Inc., una compañía de Delaware cuya oficina principal está en 1600 Amphitheatre Parkway, Mountain View (California), CA 94041, Estados Unidos ("Google"). Encuentras más información en: <https://analytics.google.com>

Doubleclick by Google: es un servicio de publicidad en línea ofrecido por Google, Inc., una compañía de Delaware cuya oficina principal está en 1600 Amphitheatre Parkway, Mountain View (California), CA 94041, Estados Unidos ("Google"). Encuentras más información en: <https://www.doubleclickbygoogle.com>

DoubleClick Clickstream: es un servicio de análisis de datos ofrecido por Doubleclick by Google, Inc., una compañía de Delaware cuya oficina principal está en 1600 Amphitheatre Parkway, Mountain View (California), CA 94041, Estados Unidos ("Google"). Encuentras más información en: <https://www.doubleclickclickstream.com>

Doubleclick: es un servicio de análisis de datos ofrecido por Doubleclick by Google, Inc., una compañía de Delaware cuya oficina principal está en 1600 Amphitheatre Parkway, Mountain View (California), CA 94041, Estados Unidos ("Google"). Encuentras más información en: <https://www.doubleclick.com>

Conservación de datos personales:

Los datos personales que proporcionas a San-Je Shop se conservarán hasta que solicites su supresión.

Exectitud y veracidad de los datos personales:

Te comprometemos a que los datos facilitados a San-Je Shop sean correctos, completos, exactos y vigentes, así como a mantenerlos debidamente actualizados.

Con el fin de garantizar la veracidad y correctitud de los datos que remitas al sitio mencionado a San-Je Shop de cualquier responsabilidad al respecto.

Aceptación y condiciones:

Como Usuario del sitio Web declaras haber sido informado de las condiciones sobre protección de datos de carácter personal, aceptas y consientes el tratamiento de los mismos por parte de San-Je Shop en la forma y para las finalidades indicadas en esta Política de Privacidad.

Revocabilidad:

Para revocar el consentimiento de los datos que envías a San-Je Shop de acuerdo a lo establecido en la legislación de protección de datos.

El ejercicio de tus derechos no incluye ningún dato que San-Je Shop esté obligado a conservar con fines administrativos, legales o de seguridad.

Cambios en la Política de Privacidad:

San-Je Shop se reserva el derecho a modificar la presente Política de Privacidad para adaptarla a novedades legislativas o jurisprudenciales, así como a prácticas de la industria.

Estas políticas estarán vigentes hasta que sean modificadas por otras debidamente publicadas.



En esta página, tenemos nuestras políticas de privacidad así como los derechos de uso.

Es un diseño simple y limpio con un leve formato.

Concurso



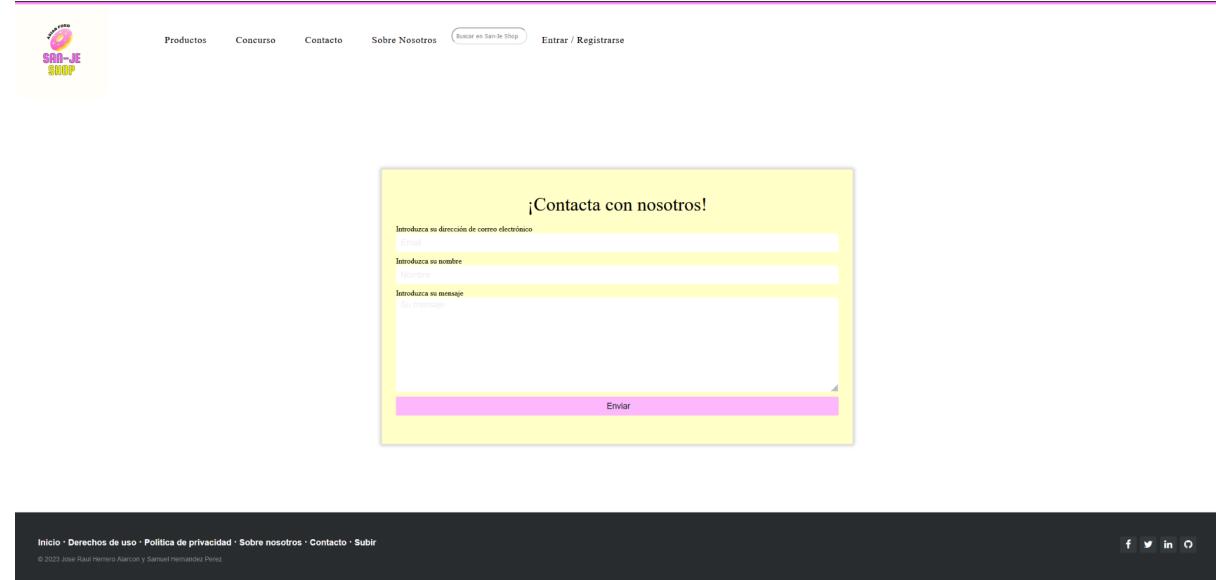
The screenshot shows a static bingo card titled "San-Je Bingo". The card is a 3x3 grid with some numbers and empty squares. The numbers are: 1, 20, 42, 79, 81 in the first row; 25, 32, 55, 65, 86 in the second row; and 4, 18, 33, 43, 66 in the third row. Below the grid, a message reads "Que la suerte te acompañe". At the top of the page, there is a navigation bar with links for Productos, Concurso, Contacto, Sobre Nosotros, a search bar, and Entrar / Registrarse.



The footer of the website includes links to Inicio, Derechos de uso, Política de privacidad, Sobre nosotros, Contacto, and Subir. It also features social media icons for Facebook, Twitter, LinkedIn, and YouTube, along with a copyright notice: © 2023 Jose Raul Herrero Alarcon y Samuel Hernandez Perez.

En el apartado de concurso, tenemos una tabla simulando un cartón de bingo, el cual de momento es estático, cosa que cambiará en un futuro, añadiendo más contenido así como haciendo cada concurso más dinámico.

Contacto



En este caso, tenemos un formulario en el centro de la página, con los colores que predominan en la página web.

Tenemos un campo en el que se debe ingresar el correo electrónico del cliente, otro campo en el que se debe ingresar el nombre del cliente que se quiere poner en contacto con nosotros, y por último un campo en el que escribirán el mensaje que quieren comunicarnos.

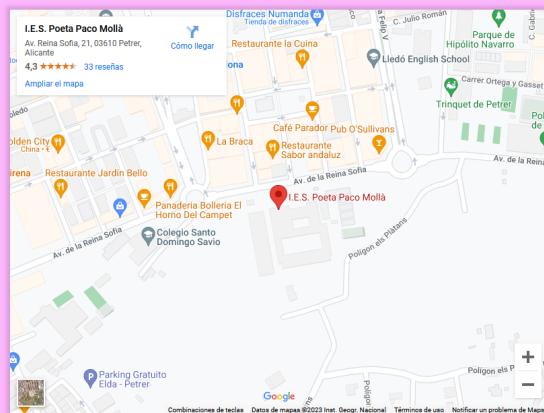
Abajo del todo tenemos un botón el cual envía los datos del formulario a nuestro correo electrónico así como mostrarte el siguiente mensaje en el cual indica que todo ha salido correctamente:

Sobre nosotros

¡Bienvenido a Sanje-Shop, tu destino para productos asiáticos únicos y difíciles de encontrar!

En Sanje-Shop, nos apasiona brindarte una experiencia culinaria excepcional y sorprendente. Nos especializamos en la venta de comida asiática, pero no cualquier comida. Nos enfocamos en ofrecerte productos auténticos y exóticos que son difíciles de encontrar en otros lugares.

Nuestro equipo se esfuerza por seleccionar cuidadosamente cada producto que ofrecemos. Trabajamos en estrecha colaboración con proveedores confiables y expertos en alimentos asiáticos para asegurarnos de que solo obtengas lo mejor de lo mejor. Nuestra pasión por la calidad y la autenticidad nos impulsa a buscar constantemente nuevos y emocionantes productos para agregar a nuestra selección.



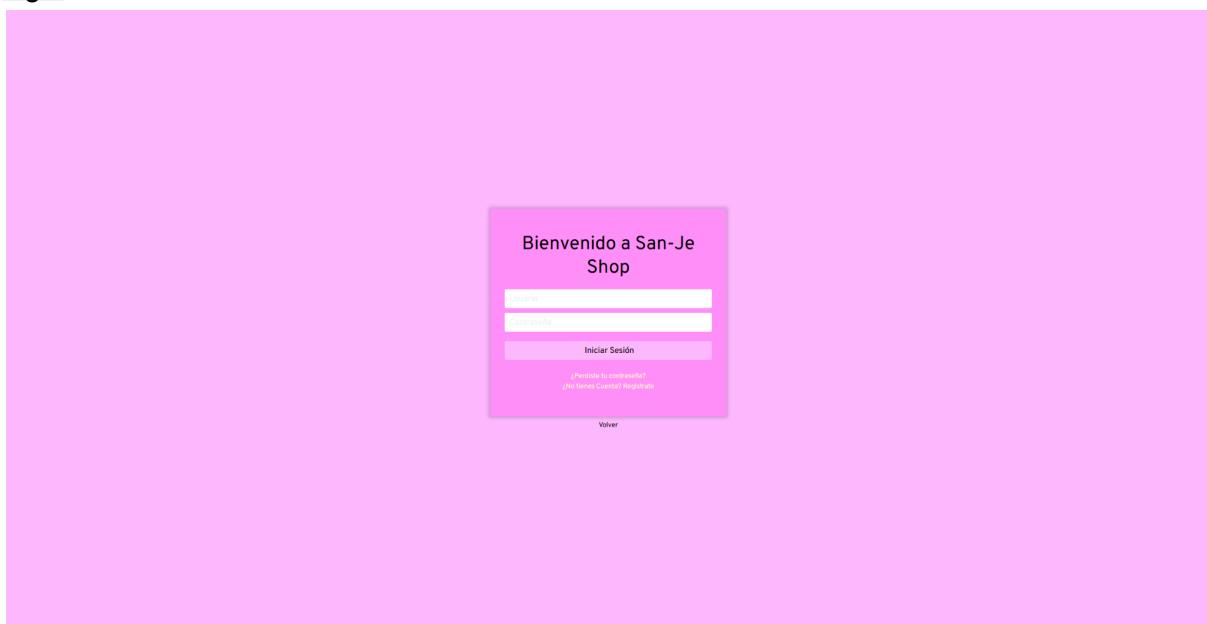
¡Te esperamos!

En este apartado, tenemos un gran cuadrado de color rosa, en el que se puede encontrar un título atractivo y una leve explicación de quienes somos y que es lo que hacemos.

Más abajo y prácticamente en el centro del cuadrado tenemos nuestra ubicación en google maps, se puede navegar por el mapa arrastrando, se puede acercar o alejar o puedes directamente hacer click en como llegar o abrir en mapa y te redirecciona a google maps.

Cerrando el cuadro con un mensaje de te esperamos que transmite buen rollo y amabilidad.

login



Aquí tenemos un formulario de inicio de sesión, en el cual el cliente tiene que indicar su nombre de usuario o correo electrónico y su contraseña, en caso de ser incorrecto, muestra el siguiente mensaje:



También tenemos tanto el botón necesario para recuperar tu contraseña, así como el de registrarte en caso de que no tengas cuenta.

Al iniciar sesión, si eres administrador, se sustituye el botón de iniciar sesión por el nombre del usuario, y en caso de ser cliente, también aparece el icono de la cesta para poder comprar.



Administrador:

[Productos](#)[Concurso](#)[Contacto](#)[Sobre Nosotros](#)[Buscar en San-Je Shop](#)[Administrador](#)

Cliente:

[Productos](#)[Concurso](#)[Contacto](#)[Sobre Nosotros](#)[Buscar en San-Je Shop](#)[Cliente](#)

Más adelante se explican las diferencias entre cliente y administrador.

productos

Saa-Je Shop

- [Productos](#)
- [Concurso](#)
- [Contacto](#)
- [Sobre Nosotros](#)
- [Entrar / Registrarse](#)

Buscar en Saa-Je Shop

(Todo) Alimentación Bebida Dulces y Snacks Postres Caja Sorpresa

Bebida Energética Godzilla Energy II
3.99€ Sin IVA

[Ver más](#)

Fanta Exotic | Maracuya, Melocotón y Naranja
1.79€ Sin IVA

[Ver más](#)

Soda Ramune Ichigo | Edición Maturi Kawaii
2.69€ Sin IVA

[Ver más](#)

Cerveza Artesana Coedo Barley | Cidre Claro
7.99€ Sin IVA

[Ver más](#)

Café Black Frio Doctor Jujutsu Kaisen
3.99€ Sin IVA

[Ver más](#)

Combinado con Yogur estilo Calpis
2.99€ Sin IVA

[Ver más](#)

3 Raciones Tallarines Frescos Tan Tan
5.18€ Sin IVA

[Ver más](#)

Fideos Kalandon con Setas Shitake
5.27€ Sin IVA

[Ver más](#)

Alga Nori Corazon Premium Gold para Sushi
3.38€ Sin IVA

[Ver más](#)

Preparado de Harina para Okonomiyaki con Kombu
8.09€ Sin IVA

[Ver más](#)

Miso Blanco Molido Grueso
5.9€ Sin IVA

[Ver más](#)

Arroz Basmati Preparado con Cevi, Chilli y Citronela
3.92€ Sin IVA

[Ver más](#)

En este apartado aparece un segundo menú de navegación para filtrar los productos por categorías, además de verse las tarjetas de los productos con su foto, su nombre, precio, y el botón de ver más el cual cambia dependiendo si eres anónimo, cliente o administrador.

Anónimo:

Bebida Energética Godzilla Energy II
3.99€ Sin IVA

[Ver más](#)

Cliente:

Bebida Energética Godzilla Energy II
3.99€ Sin IVA

[Añadir](#)

Administrador:



Bebida Energética Godzilla Energy II
3.99€ Sin IVA

Editar

En cada caso, el botón hace una cosa distinta.

articulo

Bebida Energética Godzilla Energy II



3.99€

Te traemos pura energía japonesa. La bebida energética oficial de *Godzilla*, con un diseño espectacular en el que también aparece su enemigo *Destoroyah*. Contiene 210 mg de cafeína y 680 mg de arginina por lata y un espectacular color naranja.

Necesitas iniciar sesión para añadirlo al carrito.

Iniciar Sesión

Volver

Aquí aparece tanto el nombre del artículo como su foto, su precio, su descripción y dependiendo de si eres anónimo, cliente o administrador podemos ver distintas cosas.

Anónimo:

Bebida Energética Godzilla Energy II



3.99€

Te traemos pura energía japonesa. La bebida energética oficial de *Godzilla*, con un diseño espectacular en el que también aparece su enemigo *Destoroyah*. Contiene 210 mg de cafeína y 680 mg de arginina por lata y un espectacular color naranja.

Necesitas iniciar sesión para añadirlo al carrito.

Iniciar Sesión

Volver

Tenemos un mensaje que indica que necesitas iniciar sesión para añadir al carrito, y el botón de iniciar sesión que te lleva al formulario de login.

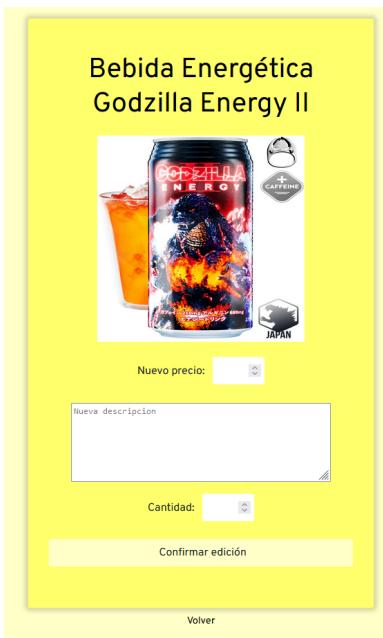
Cliente:



The screenshot shows a product page for 'Bebida Energética Godzilla Energy II'. At the top, the product name is displayed in bold black text. Below it is an image of a can of the drink next to a glass filled with an orange-colored beverage. To the right of the can are two small badges: one indicating '+ CAFFEINE' and another showing a globe with 'JAPAN' written on it. The price '3,99€' is listed below the image. A descriptive text in Spanish follows, mentioning that the drink is Japanese energy drink featuring Godzilla and Destoroyah, containing 210 mg of caffeine and 680 mg of arginine, and has an orange color. Below the text is a quantity selector set to '1' with a plus/minus button, and a large blue 'Añadir' (Add) button. At the bottom left is a 'Volver' (Return) link.

Aquí tenemos lo mismo, pero ya nos aparece el campo para seleccionar la cantidad que queremos añadir así como el botón de añadir para sumarlo al carrito.

Administrador:



The screenshot shows an edit page for the same product. The product name is at the top. Below it is the product image. A 'Nuevo precio:' field with a numeric input and a 'Nuevo descripción' text area are present. A 'Cantidad:' field with a numeric input and a 'Confirmar edición' button are at the bottom. A 'Volver' link is at the very bottom.

Aquí cambia todo y tenemos tres campos en los que podemos editar tanto el nuevo precio, como la nueva descripción y el botón de confirmar edición.

perfil

Perfil de San-Je Shop



Codigo: 11111111
Nombre: Cliente
Fecha nacimiento: 2002-02-07
Email: cliente
Telefono: 655897412

Mis Pedidos Cerrar Sesión

Volver

Aquí tenemos el perfil del cliente, mostrando su foto de perfil y sus datos, así como el botón de mis pedidos que te lleva a pedidos realizados y el botón de cerrar sesión.

Administrador:



Perfil de San-Je Shop



Codigo: 22222222
Nombre: Administrador
Fecha nacimiento: 2002-02-07
Email: admin
Telefono: 655897412

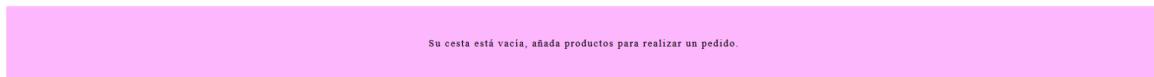
Pedidos de Sanje-Shop Cerrar Sesión

Volver

Aquí es exactamente igual, pero tenemos el botón de pedidos de Sanje-Shop, que nos permite ver todos los pedidos que han realizado los clientes. Y la foto de perfil es genérica ya que el administrador no introducido ninguna foto de perfil.

cesta

Cesta de Cliente



Nos muestra la cesta vacía con un mensaje, en caso de tener artículos la cesta nos lo muestra así:

Cesta de Cliente



Muestra la foto del producto, con el nombre, el precio unitario y las unidades en cesta. Además de un botón de eliminar, precio sin iva y precio total con iva. Acompañado de un botón para procesar el pago de dicha cesta, el cual te avisa con un mensaje en caso de haber realizado el pedido con éxito:



Pedido

Mis Pedidos

Todavía no ha realizado ningún pedido

Volver

Si no se han realizado pedidos, nos lo indica con un mensaje y los pedidos vacíos.
En caso de haber pedidos realizados, los muestra así:

Mis Pedidos

Código: 1

Fecha: 03-06-2023 18:01:08

Precio: 14,48€

Generar PDF

Generar XML

Volver

Muestra el código del pedido, la fecha de cuando se realizó y el precio que se pagó, así como los botones correspondientes para generar la factura en pdf o en xml.

PDF:

Factura nº 1

Datos del cliente:

Nº cliente: 11111111

Nombre: Cliente

E-mail: cliente

Teléfono: 655897412

Datos de la factura:

Fecha: 03-06-2023 18:01:08

Dirección: Bilbao, 6721, Calle Angosto 69

Líneas del pedido:

CÓDIGO	NOMBRE	PRECIO	I.V.A.	CANTIDAD	IMPORTE
1	Bebida Energética Godzilla Energy II	3,99 €	21%	3	14,48 €

BASE IMPONIBLE: 11,97 €

TOTAL FACTURA: 14,48 €

En caso de ser administrador, nos muestra lo siguiente.

Administrador:

Todos los Pedidos

Todavía no se ha realizado ningún pedido

Volver

Si no se han realizado pedidos, nos lo indica con un mensaje y los pedidos vacíos.
En caso de haber pedidos, los muestra así:

Todos los Pedidos

Código: 1	Cliente: Cliente	Fecha: 03-06-2023 18:01:08	Precio: 14.48€
Calle Angosto 69, 6721, Bilbao			
Volver			

Mostrando el código del pedido, el cliente al que pertenece, la fecha de cuando se realizó, el precio total del pedido y la dirección del cliente.

Factura en XML para conversión a PDF

Servlet para generarla:

```

response.setContentType("text/plain;charset=UTF-8");
out.println("<?xml version='1.0' encoding='utf-8'?>");
out.println("<factura codigo_factura=\"" + factura.getCodigo() + "\"></factura>");

out.println("<\t<fecha>");
out.println("\t<\t<dia>" + factura.getFecha().getDayOfMonth() + "</dia>");
out.println("\t<\t<mes>" + factura.getFecha().getMonthValue() + "</mes>");
out.println("\t<\t<año>" + factura.getFecha().getYear() + "</año>");
out.println("\t<\t<hora>" + factura.getFecha().getHour() + "</hora>");
out.println("\t<\t<minuto>" + factura.getFecha().getMinute() + "</minuto>");
out.println("\t<\t<segundo>" + factura.getFecha().getSecond() + "</segundo>");
out.println("<\t</fecha>");

out.println("<\t<direccion>");
out.println("\t<\t<calle>" + factura.getDireccion().getTipo() + "</calle>");
out.println("\t<\t<direccion>" + factura.getDireccion().getDireccion() + "</direccion>");
out.println("\t<\t<provincia>" + factura.getDireccion().getProvincia() + "</provincia>");
out.println("<\t</direccion>");

Pedido ped = new PedidoDAO(). getByCodigo(codigo: Integer.valueOf(request.getParameter(name: "id")));
out.println("\t<pedido_asociado codigo_pedido=\"" + ped.getCodigo() + "\"></pedido_asociado>");

out.println("\t<\t<cliente>" + ped.getDireccion().getUsuario().getNombreCompleto() + "</cliente>");
out.println("\t<\t<direccion_envio>");
out.println("\t<\t<lineas>");

ArrayList<LineaCesta> l = new CestaDAO().getLineas(nombreCesta: String.valueOf(factura.getPedido().getCodigo()), codigo: usuarioSesion.getCodigo());
double acum = 0;
for (LineaCesta linea : l) {
    out.println("\t\t<\t<producto codigo_producto=\"" + linea.getArticulo().getCodigo() + "\"></producto>");

    out.println("\t\t\t<\t<nombre>" + linea.getArticulo().getNombre() + "</nombre>");

    out.println("\t\t\t<\t<precio_unitario>" + redondear(valor: linea.getArticulo().getPvp()) + "</precio_unitario>");

    out.println("\t\t\t<\t<iva> 21% </iva>");

    out.println("\t\t\t<\t<cantidad_productos>" + linea.getCantidad() + "</cantidad_productos>");

    out.println("\t\t\t<\t<total_linea>" + redondear(valor: linea.getTotalLinea()) + "</total_linea>");

    out.println("\t\t\t<\t<total_linea_con_iva>" + redondear(linea.getTotalLinea() + linea.getTotalLinea() * 0.21) + "</total_linea_con_iva>");

    out.println("\t\t</producto>");

    acum += linea.getTotalLinea();
}

out.println("<\t</lineas>");

out.println("<\t</pedido_asociado>");

out.println("\t<base_imponible>" + redondear(valor: acum) + "</base_imponible>");

out.println("\t<total>" + redondear(acum + acum * 0.21) + "</total>");

out.println("</factura>");

response.setHeader(name: "Content-Disposition", "attachment; filename=\"Factura" + factura.getCodigo() + ".xml\"");

```

XML que genera:

```

<?xml version="1.0" encoding="utf-8"?>
<factura codigo_factura="1">
    <fecha>
        <dia>3</dia>
        <mes>6</mes>
        <año>2023</año>
        <hora>18</hora>
        <minuto>1</minuto>
        <segundo>8</segundo>
    </fecha>
    <direccion>
        <calle>Calle</calle>
        <direccion>Angosto 69</direccion>
        <provincia>Bilbao</provincia>
    </direccion>
    <pedido_asociado codigo_pedido="1">
        <cliente>Cliente</cliente>
        <direccion_envio>
            <lineas>
                <producto codigo_producto="1">
                    <nombre>Bebida Energética Godzilla Energy II</nombre>
                    <precio_unitario>3.99</precio_unitario>
                    <iva> 21% </iva>
                    <cantidad_productos>3</cantidad_productos>
                    <total_linea>11.97</total_linea>
                    <total_linea_con_iva>14.48</total_linea_con_iva>
                </producto>
            </lineas>
        </pedido_asociado>
        <base_imponible>11.97</base_imponible>
        <total>14.48</total>
    </factura>

```

Conclusiones y dificultades encontradas

Hemos tenido dificultades sobre todo con el tema de la cesta y pedido, ya que cuando hacías una cesta y procesabas el pago, se creaba un nuevo pedido en el cual entraban los datos de la cesta, y dicha cesta se borraba para poder crear posteriormente más cestas. El problema era que a la hora de imprimir el pdf desde el apartado de mis pedidos, no podíamos recuperar los datos de dicha cesta, ya que se borraban al realizar el pago.

También hemos tenido algún problema a la hora de añadir a la cesta algún artículo que ya se encontraba en la cesta, y saltaba un error que tuvimos que controlar.

A la hora de realizar el diseño de la web, hemos tenido problemas para cuadrar el submenú de los productos así como que se muestren todos los productos de manera responsive y aplicando el flex correctamente.

A la hora de realizar la base de datos, hemos tenido muchos problemas para plasmar la idea y el modelo inicial, teniendo que realizar más acciones de lo normal o incluso ir retocando y modificando el modelo entidad relación al paso para poder representar todas sus características.

Hemos tenido también problemas a la hora de convertir todos los html iniciales a jsp, ya que algunos elementos así como los estilos con el css se descuadran y hemos tenido que modificar algunas cosas para que cuadren ya que al hacerles funcionalidad, el comportamiento de dichos elementos cambiaban.

Posibles mejoras o ampliaciones

En cuanto a las mejoras, hay muchas cosas en mente que nos gustaría realizar, como por ejemplo:

- Que el tema de registrarse tenga funcionalidad, recogiendo el formulario correctamente y añadiendo los datos a la base de datos, ya que lo que tenemos por ahora tan solo es visual y no tiene ningún efecto si realizas el registro.
- Queremos darle funcionalidad al tema de la barra de búsqueda que tenemos a disposición en el menú de navegación, ya que si escribimos algo en dicha barra no realiza nada, nos gustaría que puedes buscar productos directamente ahí o incluso buscar otros apartados de la web.
- Al apartado de concurso, también queremos darle funcionalidad, haciendo que el bingo sea dinámico, pudiendo generar cartones nuevos y pudiendo tachar los números cuando le des click, así como nuevos concursos semanalmente o mensualmente, concursos como cuestionarios, bingos, cuatro en raya, ahorcados, etc... Pudiendo ganar lotes de productos o cupones de descuento para futuras compras.
- Queremos implementar la función del administrador que le permite editar un producto, que cuando inserte los nuevos datos en el formulario, los recoja correctamente la base de datos y los añada, ya que actualmente puedes escribir lo que quieras pero al pulsar el botón de confirmación no hace nada y los datos se pierden.
- Tenemos un apartado de contacto en el cual aparece un formulario que un cliente puede llenar para ponerte en contacto con nosotros, pero al igual que lo anterior, tan solo es visual y los datos no van a ningún sitio, queremos que los datos se recojan correctamente y nos llegue a nuestro correo la información escrita por el usuario.
- Nos gustaría también implementar un menú hamburguesa a la hora de hacerlo responsive, ya que lo hemos intentado pero aun así no encontrábamos una manera bonita y coherente de encajar dicho menú.
- Nos gustaría que el index (la página de bienvenida) en lugar de una foto dinámica sea un slider el cual vaya pasando mostrando distintas fotos y frases, las cuales periódicamente irían cambiando.
- Por último nos gustaría adaptar mejor nuestra web para el tema de los móviles, ya que aún teniendo modelo responsive con su flex y media queries, hay algunas relaciones de aspecto y medidas que no nos parecen las más óptimas para verlas y disfrutarlas en un dispositivo móvil.

Bibliografía

Japon-Shop → Para el tema de los productos, para algunas ideas y referencias.

Apuntes dados a lo largo del curso → Para todo el tema de funcionalidad.

Notion → Para el tema del html y css (algo de JavaScript también).

Figma → Para una idea genérica del apartado visual de nuestra página.

FreeLogo Design → Para poder crear nuestro logo personalizado.

FIN