Melyza Alejandra Rodríguez Contreras
201314821
Laboratorio de Manejo e implementación de Archivos Sección A-
10-11-2020

Manual Técnico
GTSales Marketplace

Descripción general

Este proyecto representa el concepto de un centro comercial en línea, posee clientes que pueden tomar el rol de vendedor o de comprador, exclusivamente. Se comercializan productos que son puestos a disposición del publico por medio de publicaciones hechas por los usuarios mostrando los detalles del producto en cuestión, existe la posibilidad de indicar que el producto le gusta a un usuario o por el contrario no le gusta, así mismo, los usuarios pueden contribuir denunciando alguna publicación que no sea adecuada, dicha denuncia realizada entra en revisión por parte del administrador y se determina la acción a seguir. Al momento de realizar una compra se verifica la posibilidad de adquirir los productos deseados validando le monto de créditos disponibles del usuario comprador, si todo está en orden, la compra se confirma y se le acredita el total de cada producto a su vendedor, así mismo, se le resta la totalidad de la compra al comprador.

El usuario administrador por su parte posee la capacidad de agregar nuevas categorías, revisar las denuncias realizadas por los usuarios y generar reportes que le servirán para monitorear el desempeño del sistema y poder tomar decisiones.

Descripción de las tecnologías utilizadas

- Base de datos Oracle 18c Express Edition
  Nota: La base de datos fue instalada en un contenedor Docker, versión 19.03.13

- NodeJS versión 10.19.0

- Angular versión 10.1.6

- Visual Studio Code (Como editor de texto)

API

Encriptación de Contraseña de usuario

```
/*============================ENCRIPTACION CESAR 5============================*/
function encriptar(texto, numero){
    var alfabeto = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q
    var cadena = texto.split("");
    var nuevaCadena = "";
    for(var i = 0; i < cadena.length; i++){
        for(var j = 0; j < alfabeto.length; j++){
            if(cadena[i] ==  alfabeto[j]){
                if(j + numero >= alfabeto.length){
                    var nuevoIndice = (j+numero) - alfabeto.length;
                    nuevaCadena = nuevaCadena + alfabeto[nuevoIndice];
                }else{
                    nuevaCadena = nuevaCadena + alfabeto[j+numero];
                }
                break;
            }
        }
    }
    return nuevaCadena;
}
```

Login de usuario

```
/*==================================LOGIN==================================*/
router.get('/login', function(request, response){
    console.log("Entrando a login");
    var email = request.query.email;
    var password = request.query.password;
    var passwordEncriptado = encriptar(password,5);
    sql = ` SELECT *
            FROM usuario
            WHERE
                    email = :email
                AND
                    contrasena = :password`;
    dao.open(sql, [email, passwordEncriptado], false, response);
    response.end;

});
```

Registro de usuarios

```
*==============================REGISTRO================================*/
outer.get('/registro', function(request, response){
   console.log("Entrando a registro");
   var usuario = 0;
   var nombre = request.query.nombre;
   var email = request.query.email;
   var password = request.query.password;
   var passwordEncriptado = encriptar(password,5);
   var fecha = request.query.fecha;
   var foto = request.query.foto;
   var pais = request.query.pais;
   var creditos = 10000;
   var tipo = 2;
   var estado = 0;
   sql = ` INSERT INTO usuario
           (usuario, nombre, email, contrasena, pais, foto, creditos, tipo_usuario, estado, fecha_nacimien
           VALUES
           (:usuario, :nombre, :email, :password, :pais, :foto, :creditos, :tipo, :estado, TO_DATE(:fecha,
   dao.open(sql, [usuario, nombre, email, passwordEncriptado, pais, foto, creditos, tipo, estado, fecha],
   response.end;
```

Confirmación de cuenta de usuario

```
/*==================================CONFIRMAR=================================*/
router.get('/confirmarcuenta', function(request, response){
    console.log("Entrando a confirmarcuenta");
    sql = ` UPDATE usuario
            SET estado = 1
            WHERE email = :email`;
    var email = request.query.email;
    dao.open(sql, [email], true, response);
    response.end;

});
```

Ver todos los productos que no pertenecen al usuario

```
/*============================VER TODOS LOS PRODUCTOS==============================*/
router.get('/productos', function(request, response){
    console.log("Entrando a ver productos");
    var usuario = request.query.usuario;
    sql = ` SELECT *
            FROM producto
            WHERE
                    propietario != :usuario AND estado = 1`;
    dao.open(sql, [usuario], false, response);
    response.end;
});
```

Ver todos los productos que pertenecen al usuario

```
/*============================VER TODOS MIS PRODUCTOS============================*/
router.get('/misproductos', function(request, response){
    console.log("Entrando a ver productos mios");
    var usuario = request.query.usuario;
    sql = ` SELECT *
            FROM producto
            WHERE
                    propietario = :usuario AND estado = 1`;
    dao.open(sql, [usuario], false, response);
    response.end;
});
```

Ver todas las categorías existentes

```
/*============================VER CATEGORIAS============================*/
router.get('/categorias', function(request, response){
    console.log("Entrando a ver categorias");
    sql = ` SELECT *
            FROM categoria`;
    dao.open(sql, [], false, response);
    response.end;
});
```

Insertar una nueva categoría

```
/*==============================NUEVA CATEGORIA================================*/
router.get('/nuevacategoria', function(request, response){
    console.log("Entrando a nuevacategoria");
    sql = ` INSERT INTO categoria(categoria, nombre)
            VALUES (0, :categoria)`;
    var categoria = request.query.categoria;
    dao.open(sql, [categoria], true, response);
    response.end;
});
```

Búsqueda de productos por criterio de búsqueda

```
/*==================BUSQUEDA DE PRODUCTOS POR CRITERIO====================*/
router.get('/busquedaproductos', function(request, response){
    console.log("Entrando A BUSCAR PRODUCTOS");
    sql = `(
            (SELECT * FROM producto WHERE NOMBRE LIKE :criterio AND propietario != :usuario AND est
            UNION
            (SELECT * FROM producto WHERE DETALLE LIKE :criterio AND propietario != :usuario AND es
        )
        UNION
        (SELECT * FROM producto WHERE PALABRAS_CLAVE LIKE :criterio AND propietario != :usuario AND
    var criterio = request.query.criterio;
    var usuario = parseInt(request.query.usuario);
    dao.open(sql, [criterio, usuario], false, response);
    response.end;
});
```

Búsqueda de productos por categoría

```
/*==================BUSQUEDA DE PRODUCTOS POR CATEGORIA====================*/
router.get('/busquedacategorias', function(request, response){
    console.log("Entrando A BUSCAR PRODUCTOS por categoria");
    sql = ` SELECT *
            FROM producto
            WHERE categoria = :categoria AND propietario != :usuario AND estado = 1`;
    var categoria = request.query.categoria;
    var usuario = parseInt(request.query.usuario);
    dao.open(sql, [categoria, usuario], false, response);
    response.end;
});
```

Productos por orden de precio viendo el mayor precio primero

```
/*==================BUSQUEDA DE PRODUCTOS POR MAYOR PRECIO=====================*/
router.get('/busquedamayor', function(request, response){
    console.log("Entrando A BUSCAR PRODUCTOS por mayor precio");
    sql = ` SELECT *
            FROM producto
            WHERE propietario != :usuario AND estado = 1
            ORDER BY precio DESC`;
    var usuario = parseInt(request.query.usuario);
    dao.open(sql, [usuario], false, response);
    response.end;
});
```

Productos por orden de precio viendo el menor precio primero

```
/*==================BUSQUEDA DE PRODUCTOS POR MENOR PRECIO=====================*/
router.get('/busquedamenor', function(request, response){
    console.log("Entrando A BUSCAR PRODUCTOS por menor precio");
    sql = ` SELECT *
            FROM producto
            WHERE propietario != :usuario AND estado = 1
            ORDER BY precio ASC`;
    var usuario = parseInt(request.query.usuario);
    dao.open(sql, [usuario], false, response);
    response.end;
});
```

Insertar un nuevo producto

```
/*=========================NUEVO PRODUCTO============================*/
router.get('/nuevoproducto', function(request, response){
    console.log("Entrando a nuevo producto");
    sql = ` INSERT INTO producto(producto, nombre, detalle, palabras_clave, precio, categoria, propietar
            VALUES (0, :nombre, :detalle, :palabras_clave, :precio, :categoria, :usuario, :foto)`;
    var nombre = request.query.nombre;
    var detalle = request.query.detalle;
    var palabras_clave = request.query.palabras_clave;
    var precio = request.query.precio;
    var categoria = request.query.categoria;
    var usuario = request.query.usuario;
    var foto = request.query.foto;
    dao.open(sql, [nombre, detalle, palabras_clave, precio, categoria, usuario, foto], true, response);
    response.end;
});
```

Ver el producto actual

```
/*=========================VER PRODUCTO ACTUAL=========================*/
router.get('/productoactual', function(request, response){
    console.log("Entrando a ver un solo producto");
    sql = ` SELECT p.producto, p.nombre, p.detalle, p.palabras_clave, p.precio, p.categoria, p.propietar
            FROM producto p, categoria c
            WHERE producto = :producto AND c.categoria = p.categoria`;
    var producto = request.query.producto;
    dao.open(sql, [producto], false, response);
    response.end;
});
```

Ver like de un producto

```
/*=================================VER LIKE=================================*/
router.get('/verlike', function(request, response){
    console.log("Entrando a ver si le dio like");
    sql = ` SELECT *
            FROM megusta WHERE producto = :producto AND usuario = :usuario`;
    var producto = request.query.producto;
    var usuario = request.query.usuario;
    dao.open(sql, [producto, usuario], false, response);
    response.end;
});
```

Dar like a un producto

```
/*=================================DAR LIKE=================================*/
router.get('/darlike', function(request, response){
    console.log("Entrando a dar like inicial");
    sql = ` INSERT INTO megusta(id_megusta, producto, usuario, estado)
            VALUES (0, :producto, :usuario, :estado)`;
    var usuario = request.query.usuario;
    var producto = request.query.producto
    var estado = request.query.estado;

    dao.open(sql, [producto, usuario, estado], true, response);
    response.end;
});
```

Actualizar el estado de like

```
/*==============================ACTUALIZAR LIKE================================*/
router.get('/actualizarlike', function(request, response){
    console.log("Entrando a actualizar like");
    sql = ` UPDATE  megusta SET estado=:estado
            WHERE producto=:producto AND usuario=:usuario`;
    var usuario = request.query.usuario;
    var producto = request.query.producto
    var estado = request.query.estado;

    dao.open(sql, [estado, producto, usuario], true, response);
    response.end;
});
```

Realizar un comentario sobre un producto

```
/*==============================COMENTAR================================*/
router.get('/comentar', function(request, response){
    console.log("Entrando a comentar");
    sql = ` INSERT INTO comentario(comentario, usuario, producto, contenido, fecha)
            VALUES(0, :usuario, :producto, :contenido, TO_DATE(:fecha,'DD/MM/YYYY'))`;
    var usuario = request.query.usuario;
    var producto = request.query.producto
    var contenido = request.query.contenido;
    var fecha =  request.query.fecha;

    dao.open(sql, [usuario, producto, contenido, fecha], true, response);
    response.end;
});
```

Obtener todos los comentarios de un producto

```
/*==============================VER COMENTARIOS================================*/
router.get('/vercomentarios', function(request, response){
    console.log("Entrando a ver comentarios");
    sql = ` SELECT c.COMENTARIO, c.USUARIO, c.PRODUCTO, c.CONTENIDO, c.FECHA, u.NOMBRE, u.FOTO
            FROM comentario c, usuario u
            WHERE c.USUARIO = u.USUARIO AND c.PRODUCTO = :producto
            ORDER BY c.comentario DESC`;
    var producto = request.query.producto;

    dao.open(sql, [producto], false, response);
    response.end;
});
```

Actualizar la información de un usuario

```
/*=========================ACTUALIZAR UN USUARIO============================*/
router.get('/actualizarusuario', function(request, response){
    console.log("Entrando a actualizar usuario");

    sql = ` UPDATE usuario
            SET nombre = :nombre, contrasena = :contrasena, pais = :pais, foto = :foto
            WHERE usuario = :usuario`;
    var nombre = request.query.nombre;
    var contrasena = request.query.contrasena;
    var pais = request.query.pais;
    var foto = request.query.foto;
    var usuario = request.query.usuario;
    var fecha = request.query.fecha;
    var cont = request.query.cont;
    console.log(fecha);

    if(cont == 1){
        var contrasenaEncriptada = encriptar(contrasena, 5);
        contrasena = contrasenaEncriptada;
        dao.open(sql, [nombre, contrasena, pais, foto, usuario], true, response);
    }else if(cont == 0){
        dao.open(sql, [nombre, contrasena, pais, foto, usuario], true, response);
    }

    response.end;
});
```

Obtener los datos de un usuario

```
/*================================OBTENER USUARIO=================================*/
router.get('/obtenerusuario', function(request, response){
    console.log("Entrando a obtener usuario");
    sql = ` SELECT *
            FROM usuario
            WHERE usuario = :usuario`;
    var usuario = request.query.usuario;

    dao.open(sql, [usuario], false, response);
    response.end;
});
```

Realizar una denuncia

```
/*==============================DENUNCIAR PRODUCTO================================*/
router.get('/denunciar', function(request, response){
    console.log("Entrando a denunciar producto");
    sql = ` INSERT INTO denuncia
            (denuncia, usuario, producto, contenido, estado, fecha)
            VALUES (0, :usuario, :producto, :contenido, 0, TO_DATE(:fecha,'DD/MM/YYYY'))`;
    var usuario = request.query.usuario;
    var producto = request.query.producto;
    var contenido = request.query.contenido;
    var fecha = request.query.fecha;

    dao.open(sql, [usuario, producto, contenido, fecha], true, response);
    response.end;
});
```

Obtener todas las denuncias

```
/*==============================VER DENUNCIAS================================*/
router.get('/verdenuncias', function(request, response){
    /*Solo las denuncias que no han sido revisadas*/
    console.log("Entrando a ver denuncias");
    sql = ` SELECT d.denuncia, d.usuario, d.producto, d.contenido, u.nombre, u.foto, p.nombre, p.foto, p
            FROM denuncia d, usuario u, producto p
            WHERE d.estado = 0
            AND p.producto = d.producto
            AND u.usuario = d.usuario
            ORDER BY d.fecha ASC`;
    dao.open(sql, [], false, response);
    response.end;
});
```

Denegar una denuncia

```
/*==============================DENEGAR DENUNCIA================================*/
router.get('/denegardenuncia', function(request, response){
    console.log("Entrando a denegar denuncia");
    sql = ` UPDATE denuncia
            SET estado = 2
            WHERE denuncia = :denuncia`;
    var denuncia = request.query.denuncia;
    dao.open(sql, [denuncia], true, response);
    response.end;
});
```

Aceptar una denuncia

```
/*==============================ACEPTAR DENUNCIA==================================*/
router.get('/aceptardenuncia', function(request, response){
    console.log("Entrando a aceptar denuncia");
    sql = ` UPDATE denuncia
            SET estado = 1
            WHERE denuncia = :denuncia`;
    var denuncia = request.query.denuncia;
    dao.open(sql, [denuncia], true, response);
    response.end;
});

router.get('/cambiarestadop', function(request, response){
    console.log("Entrando a pner producto en estado 2 para que no se muestre mas");
    sql = ` UPDATE producto
            SET estado = 2
            WHERE producto = :producto`;
    var producto = request.query.producto;
    dao.open(sql, [producto], true, response);
    response.end;
});
```

Obtener todos los paises registrados

```
/*==============================VER TODOS LOS PAISES==================================*/
router.get('/paises', function(request, response){
    sql = "SELECT * FROM pais";
    dao.open(sql, [], false, response);
    response.end;
});
```

Insertar una categoría

```
/*==============================INSERTAR UNA CATEGORIA==================================*/
router.get('/nuevaCategoria', function(request, response){
    sql = "INSERT INTO categoria(CATEGORIA, NOMBRE) values (:categoria, :nombre)";
    var categoria = parseInt(request.query.categoria);
    var nombre = request.query.nombre;
    dao.open(sql, [categoria, nombre], true, response);
    response.end;
});
```

Insertar una acción a la bitácora

```
/*==============================INSERTAR BITACORA================================*/
router.get('/insertarbitacora', function(request, response){
    sql = ` INSERT INTO bitacora
            (bitacora, accion, fecha, correo)
            VALUES(0, :accion, TO_DATE(:fecha, 'DD/MM/YYYY'), :correo)`;
    var accion = request.query.accion;
    var fecha = request.query.fecha;
    var correo = request.query.correo;
    dao.open(sql, [accion, fecha, correo], true, response);
    response.end;
});
```

Reporte de bitácora

```
/*==============================VER BITACORA================================*/
router.get('/verbitacora', function(request, response){
    console.log("Entrando a ver bitacora");
    sql = ` SELECT *
            FROM bitacora
            ORDER BY fecha DESC`;
    dao.open(sql, [], false, response);
    response.end;
});
```

Reporte de 10 productos más vendidos

```
/*==============================10+ Vendidos================================*/
router.get('/diezmasvendidos', function(request, response){
    console.log("Entrando a ver diez mas vendidos");
    sql = `
            SELECT p.producto, p.nombre, datos.cantidad, u.usuario, u.nombre
            FROM
                (   SELECT producto AS producto, SUM(cantidad) AS cantidad
                    FROM detalle
                    GROUP BY producto
                ) datos, producto p, usuario u
            WHERE u.usuario = p.propietario AND p.producto = datos.producto AND ROWNUM <=10
            ORDER BY datos.cantidad DESC
    `;
    dao.open(sql, [], false, response);
    response.end;
});
```

Reporte de 10 productos con más me gusta

```
/*================================10+ MEGUSTA=================================*/
router.get('/diezmasmegusta', function(request, response){
    console.log("Entrando a ver diez mas megusta");
    sql = `
                SELECT p.producto, p.nombre, datos.cantidad, u.usuario, u.nombre
                FROM
                    (   SELECT producto AS producto, COUNT(*) AS cantidad
                        FROM megusta
                        WHERE estado = 1
                        GROUP BY producto
                    ) datos, producto p, usuario u
                WHERE u.usuario = p.propietario AND p.producto = datos.producto AND ROWNUM <=10
                ORDER BY datos.cantidad DESC
    `;
    dao.open(sql, [], false, response);
    response.end;
});
```

Reporte de 10 productos con más no me gusta

```
/*================================10+ NO ME GUSTA================================*/
router.get('/diezmasnomegusta', function(request, response){
    console.log("Entrando a ver diez mas no megusta");
    sql = `
                SELECT p.producto, p.nombre, datos.cantidad, u.usuario, u.nombre
                FROM
                    (   SELECT producto AS producto, COUNT(*) AS cantidad
                        FROM megusta
                        WHERE estado = 2
                        GROUP BY producto
                    ) datos, producto p, usuario u
                WHERE u.usuario = p.propietario AND p.producto = datos.producto AND ROWNUM <=10
                ORDER BY datos.cantidad DESC
    `;
    dao.open(sql, [], false, response);
    response.end;
});
```

Clientes con más y menos créditos

```
router.get('/diezclientescreditos', function(request, response){
    console.log("Entrando a ver diez mas y menos creditos");
    sql = `
                SELECT DISTINCT *
                FROM usuario
                WHERE tipo_usuario = 2 AND ROWNUM <= 10
                ORDER BY creditos DESC

        `;
    dao.open(sql, [], false, response);
    response.end;
});

router.get('/diezclientescreditosmenos', function(request, response){
    console.log("Entrando a ver diez mas y menos creditos");
    sql = `
                SELECT DISTINCT *
                FROM usuario
                WHERE tipo_usuario = 2 AND ROWNUM <= 10
                ORDER BY creditos ASC

        `;
    dao.open(sql, [], false, response);
    response.end;
});
```

Reporte de Clientes que han realizado denuncias

```
/*================================10+ MAS DENUNCIAS=================================*/
router.get('/diezmasdenuncias', function(request, response){
    console.log("Entrando a ver diez mas denuncias");
    sql = `
            SELECT u.usuario, u.nombre, u.email, u.fecha_nacimiento, datos.cantidad
            FROM
                (   SELECT usuario AS usuario, COUNT(*) AS cantidad
                    FROM denuncia
                    GROUP BY usuario
                ) datos, usuario u
            WHERE u.usuario = datos.usuario AND ROWNUM <=10
            ORDER BY datos.cantidad DESC
    `;
    dao.open(sql, [], false, response);
    response.end;
});
```

Reporte de Clientes que han publicado más productos

```
/*================================10+ Vendidos=====================================*/
router.get('/diezmasproducto', function(request, response){
    console.log("Entrando a ver diez mas productos publicados");
    sql = `
                SELECT u.usuario, u.nombre, datos.cantidad, u.email, u.fecha_nacimiento
                FROM
                    (   SELECT propietario AS usuario, COUNT(*) AS cantidad
                        FROM producto
                        GROUP BY propietario
                    ) datos, usuario u
                WHERE u.usuario = datos.usuario AND ROWNUM <=10
                ORDER BY datos.cantidad DESC
    `;
    dao.open(sql, [], false, response);
    response.end;
});
```

Reporte de créditos, usuarios y productos de cada país

```
/*================================Info paises=====================================*/
router.get('/infopaises', function(request, response){
    console.log("Entrando a ver info paises");
    sql =   `
                SELECT datos2.pais, datos2.creditos, datos2.productos, COUNT(u.usuario) AS usuarios, p.no
                FROM
                (
                    SELECT datos.pais AS pais, datos.creditos AS creditos, COUNT(p.producto) AS productos
                    FROM
                    (
                        SELECT pais AS pais, SUM(creditos) AS creditos
                        FROM usuario
                        GROUP BY pais
                    ) datos, usuario u, producto p
                    WHERE p.propietario = u.usuario AND u.pais = datos.pais
                    GROUP BY datos.pais, datos.creditos
                )datos2, usuario u, pais p
                WHERE u.pais = datos2.pais AND u.pais = p.pais
                GROUP BY datos2.pais, datos2.creditos, datos2.productos, p.nombre
                ORDER BY datos2.creditos DESC
            `;
    dao.open(sql, [], false, response);
    response.end;
});
```

Añadir un producto al carrito de compras

```
/*==============================ANADIR CARRITO==================================*/
router.get('/anadircarrito', function(request, response){
    console.log("Entrando a anadircarrito");
    sql = ` INSERT INTO carrito
            (carrito, usuario, producto, cantidad, estado)
            VALUES (0, :usuario, :producto, :cantidad, 1)`;
    var usuario = request.query.usuario;
    var producto = request.query.producto;
    var cantidad = request.query.cantidad;
    dao.open(sql, [usuario, producto, cantidad], true, response);
    response.end;
});
```

Ver todos los productos en el carrito de compra de un cliente

```
/*==============================VER CARRITO==================================*/
router.get('/vercarrito', function(request, response){
    console.log("Entrando a ver carrito");
    sql = ` SELECT c.carrito, c.producto, c.usuario, c.cantidad, c.estado, p.nombre, p.precio, p.pr
            FROM carrito c, producto p, usuario u
            WHERE c.usuario = :usuario AND c.estado = 1 AND p.producto = c.producto AND p.propietar
    var usuario = request.query.usuario;
    dao.open(sql, [usuario], false, response);
    response.end;

});
```

Vaciar el carrito de compras de un cliente

```
/*==============================VACIAR CARRITO==================================*/
router.get('/vaciarcarrito', function(request, response){
    console.log("Entrando a vaciar carrito");
    sql = ` UPDATE carrito
            SET estado = 0
            WHERE usuario = :usuario`;
    var usuario = request.query.usuario;
    dao.open(sql, [usuario], true, response);
    response.end;
});
```

Quitar un producto del carrito de compras

```
/*================================QUITAR CARRITO=================================*/
router.get('/quitarCarrito', function(request, response){
    console.log("Entrando a quitar un solo item del carrito");
    sql = ` UPDATE carrito
            SET estado = 0
            WHERE carrito = :carrito`;
    var carrito = request.query.carrito;
    dao.open(sql, [carrito], true, response);
    response.end;
});
```

Recuperación de contraseña

```
/*================================RECUPERAR CONTRASENA=================================*/
router.get('/recuperarcontrasena', function(request, response){
    console.log("Entrando a recuperar contrasena");
    sql = ` SELECT *
            FROM usuario
            WHERE email = :correo`;
    var correo = request.query.correo;
    dao.open(sql, [correo], false, response);
    response.end;
});

router.get('/setnuevacontrasena', function(request, response){
    console.log("Entrando a setnuevacontra contrasena");
    sql = ` UPDATE usuario
            SET contrasena = :contrasena
            WHERE email = :correo`;
    var contrasena = request.query.contrasena;
    var contrEncriptada = encriptar(contrasena, 5);
    var correo = request.query.correo;
    dao.open(sql, [contrEncriptada, correo], true, response);
    response.end;
});
```

Manejo de imágenes

```
/*============================MANDAR IMAGEN A SERVIDOR=============================*/
const fs = require('fs');
const { response } = require('express');
app.post('/upload', (req, res) =>{
    const file = req.body.file;
    const name = req.body.name;
    const binaryData = new Buffer(file.replace(/^data:image\/jpeg;base64,/,""), "base64").toString("binary"
    fs.writeFile("/home/melyza/Desktop/serverImagenes/"+name, binaryData, "binary", (err) =>{
        console.log(err);
    })
    res.json(1);
});
```

Envío de correos electrónicos

```
/*============================CORREOS ELECTRONICOS================================*/
app.post('/enviarcorreo', (req, res)=>{
    var transporter = nodemailer.createTransport({…
    var mailOptions = {
        from: "GTSales Marketplace",
        to: req.body.destinatario,
        subject: req.body.asunto,
        html: req.body.texto
    }
    transporter.sendMail(mailOptions, (err, info)=>{
        if(err){
           res.status(500).send(err.message);
        }else{
            console.log("Correo enviado");
            res.status(200).jsonp(req.body);
        }
    });
});
```

Actualizar créditos de un usuario

```
/*============================ACTUALIZAR CREDITOS================================*/
router.get('/actualizarcreditos', function(request, response){
    console.log("Entrando a actualizarcreditos");
    sql = ` SELECT creditos
            FROM usuario
            WHERE usuario = :usuario`;
    var usuario = request.query.usuario;
    dao.open(sql, [usuario], false, response);
    response.end;
});
```

Crear una factura

```
/*============================CREAR FACTURA================================*/
router.get('/crearfactura', function(request, response){
    console.log("Entrando a crearfactura");
    sql = ` INSERT INTO factura
            (correlativo, usuario, fecha)
            VALUES
            (0, :usuario, TO_DATE(:fecha, 'DD/MM/YYYY'))`;
    var usuario = request.query.usuario;
    var fecha = request.query.fecha;
    dao.open(sql, [usuario, fecha], true, response);
    response.end;
});
```

Ver facturas

```
router.get('/verfacturas', function(request, response){
    console.log("Entrando a ver facturas");
    sql = ` SELECT *
            FROM factura
            WHERE usuario = :usuario
            ORDER BY correlativo DESC`;
    var usuario = request.query.usuario;
    dao.open(sql, [usuario], false, response);
    response.end;
});
```

Insertar detalle de factura

```
router.get('/insertardetalle', function(request, response){
    console.log("Entrando a insertar detalle");
    sql = ` INSERT INTO detalle
            (id_detalle, correlativo, producto, cantidad)
            VALUES
            (0, :correlativo, :producto, :cantidad)`;
    var correlativo = request.query.correlativo;
    var producto = request.query.producto;
    var cantidad = request.query.cantidad;
    dao.open(sql, [correlativo, producto, cantidad], true, response);
    response.end;
});
```

Cambiar estado de un producto en el carrito de compras de un usuario

```
router.get('/cambiarestadocarrito', function(request, response){
    console.log("Entrando a quitarcarrito");
    sql = ` UPDATE carrito
            SET estado = 0
            WHERE carrito = :carrito`;
    var carrito = request.query.carrito;
    dao.open(sql, [carrito], true, response);
    response.end;
});
```

Actualizar créditos post-compra

```
router.get('/actualizarmiscreditos', function(request, response){
    console.log("Entrando a actualizarmiscreditos");
    sql = ` UPDATE usuario
            SET creditos = :creditos
            WHERE usuario = :usuario`;
    var creditos = request.query.creditos;
    var usuario = request.query.usuario;
    dao.open(sql, [creditos, usuario], true, response);
    response.end;
});

router.get('/actualizarcreditosvendedor', function(request, response){
    console.log("Entrando a actualizarcreditosvendedor");
    sql = ` UPDATE usuario
            SET creditos = creditos  + :creditos
            WHERE usuario = :usuario`;
    var creditos = request.query.creditos;
    var usuario = request.query.usuario;
    dao.open(sql, [creditos, usuario], true, response);
    response.end;
});
```

Aplicación Web

Resumen de componentes

Componente	Descripción
Carrito	Carrito de compras del usuario **quitarProducto(carrito)** -Quita el producto indicado del carrito de compra del usuario **realizarCompra()** -Confirma la compra de todos los productos del carrito **vaciarCarrito()** -Quita todos los productos actuales del carrito de compras
Categorias	Manejo de categorías por parte del administrador **insertarCategoria()** -Insertar una nueva categoría
Confirmarcuenta	Confirmación de cuenta por parte del usuario **confirmar()** -Confirmación de cuenta por parte del usuario, ingreso de contraseña

Denunciarproducto	Pantalla para realizar la denuncia de un producto **realizarDenuncia()** -Realizar denuncia por parte de un usuario, ingreso del motivo para realizar la denuncia.
Denuncias	Visualización de denuncias por parte del usuario administrador **denegarDenuncia()** -Denegar la denuncia sin realizar ningún cambio **eliminarProducto()** -Aceptar la denuncia y cambiar el estado del producto a no visible para los usuarios
Error-pagina	Pagina que se muestra cuando se intenta acceder mientras no se ha realizado login.
Login	Pantalla de login para iniciar sesión por parte del usuario **onSaveForm()** -Envia los datos del usuario para realizar el login. **IrARecuperar()** -Redirige al usuario a la pantalla de recuperar contraseña.
Logout	Pantalla de fin de sesión.
Misproductos	Pantalla de visualización de productos que han sido publicados por el usuario.
Nuevo	Registro de un nuevo producto por parte del usuario. **Registrar()** -Envia los datos del producto para ser registrado.
Perfil	Presentación del perfil del usuario **actualizar()** -Actualización de datos de usuario
Productoactual	Pagina donde se visualiza el producto actual, sus detalles y todas las acciones que se pueden aplicar. **buscarProductos()** -Realiza una búsqueda de productos por palabras clave **buscarProductosCategoria()** -Realiza una búsqueda de productos por categoría **buscarProductosMayor()** -Ordena los productos mostrando primero el precio mayor

	buscarProductosMenor() -Ordena los productos mostrando primero el precio menor
Recuperar	Solicitud del correo del usuario para recuperar su contraseña **recuperar()** -Envío de correo electrónico para recuperar contraseña
Registro	Formulario de registro de usuario **registrarse()** -Envío de datos del usuario a registrar **resetCampos()** -Borra los campos que el usuario ha llenado
Reportes	Generación de reportes para vista del administrador **verBitacoraASC()** **verBitacoraDESC()** **verDiezMasVendidos()** **verDiezMasMeGusta()** **verDiezMasNoMeGusta()** **verDiezClientesCreditos()** **verDiezClientesCreditosMenos()** **verDiezClientesMasDenuncias()** **verDiezClientesMasProductos()** **verPaisesReportes()** **verReporte()**
Tarjeta	Visualización de página de bienvenida del usuario **irCliente()** -Ir a perfil de cliente **irAdmin()** -Ir a perfil de administrador