

# SISTEMA

---

## Índice

### Contenido

INTRODUCCIÓN	2
REQUISITOS	2
DESARROLLO	4

## INTRODUCCIÓN

La empresa Burgers SRL desea implementar un sitio web para recibir pedidos en modalidad takeaway. Una persona realiza la compra desde la página web, eligiendo los productos, seleccionando la sucursal, y abonando en efectivo en el local al retirar el pedido u online a través de una pasarela de pago (mercadopago).

Desde un panel administrativo los vendedores podrán visualizar los pedidos entrantes con el estado “Pendiente” y modificar los mismos con los estados “En preparación”, “Listo para retirar”, “Entregado” y “Cancelado”

## REQUISITOS

### Requisitos sitio web institucional

El sitio web debe mostrar un catálogo de productos, con los datos: **título, descripción, precio, input de cantidad, botón para agregar en el carrito.**

El sitio web debe mostrar un listado de las **sucursales con sus domicilios, números de teléfono y un link a la dirección en google maps.**

El sitio web debe permitir a un cliente registrarse, con los datos: **nombre, apellido, celular, email, contraseña.**

El sitio web debe permitir a un cliente registrado iniciar sesión con los datos: **email, contraseña.**

El sitio web debe permitir a una persona logueada agregar 1 o varios productos al carrito.

El carrito de compras debe permitir seleccionar la sucursal de retiro, medio de pago, eliminar el producto del carrito.

El sitio web debe permitir a una persona postularse para trabajar para la empresa: **nombre y apellido, email, whatsapp, input para adjuntar CV y botón de Enviar.** Luego de enviados los datos deberá mostrar una pantalla que indique que pronto será contactado por la empresa.

### Requisitos para el panel administrativo

El sistema debe permitir gestionar **clientes**, con los datos: **nombre, dni, email y dni.**

El sistema debe permitir gestionar **productos**, con los datos: Tipo de productos

El sistema debe permitir gestionar **pedidos**, con los datos:

El sistema debe permitir gestionar **sucursales**, con los datos: **Localidad ,domicilio, celular**

El sistema debe permitir gestionar **postulaciones**, con los datos: **botón de ver postulación, nombre y apellido, whatsapp, correo y link para descargar CV.**

El sistema debe permitir gestionar usuarios y permisos.

### Páginas de la competencia

URL	Alumno
<a href="https://www.mostazaweb.com.ar/">https://www.mostazaweb.com.ar/</a>	Ezequiel
<a href="https://linktr.ee/mbhargentina">https://linktr.ee/mbhargentina</a>	Maricel
<a href="https://www.thefoodtruckstore.com.ar/">https://www.thefoodtruckstore.com.ar/</a>	Tomas
<a href="http://www.theburgercompany.com.ar/">http://www.theburgercompany.com.ar/</a>	Anderson

### Redes sociales

No tiene.

### Plantilla para la web (bootstrap)

Buscar en Internet plantillas compatibles con bootstrap, se deja estos recursos de ejemplo:

<https://themewagon.com/theme-framework/bootstrap-5/>

<https://bootstrapmade.com/>

URL	Alumno
<a href="https://www.smarteyeapps.com/free_demo/fast-food-restaurant-website-template-free-download">https://www.smarteyeapps.com/free_demo/fast-food-restaurant-website-template-free-download</a>	maricel, tomas, ezequiel, anderson

## DESARROLLO

### Tablas del sistema

clientes (idcliente, nombre, apellido, telefono, correo, clave) **anderson**

productos (idproducto, nombre, descripcion, imagen, precio) **maricel**

sucursales (idsucursal, nombre, domicilio, telefono, link\_mapa) **ezequiel**

pedidos(idpedido, fk\_idsucursal, fk\_idcliente, fecha, fk\_idestado, total, comentario) **tomas**

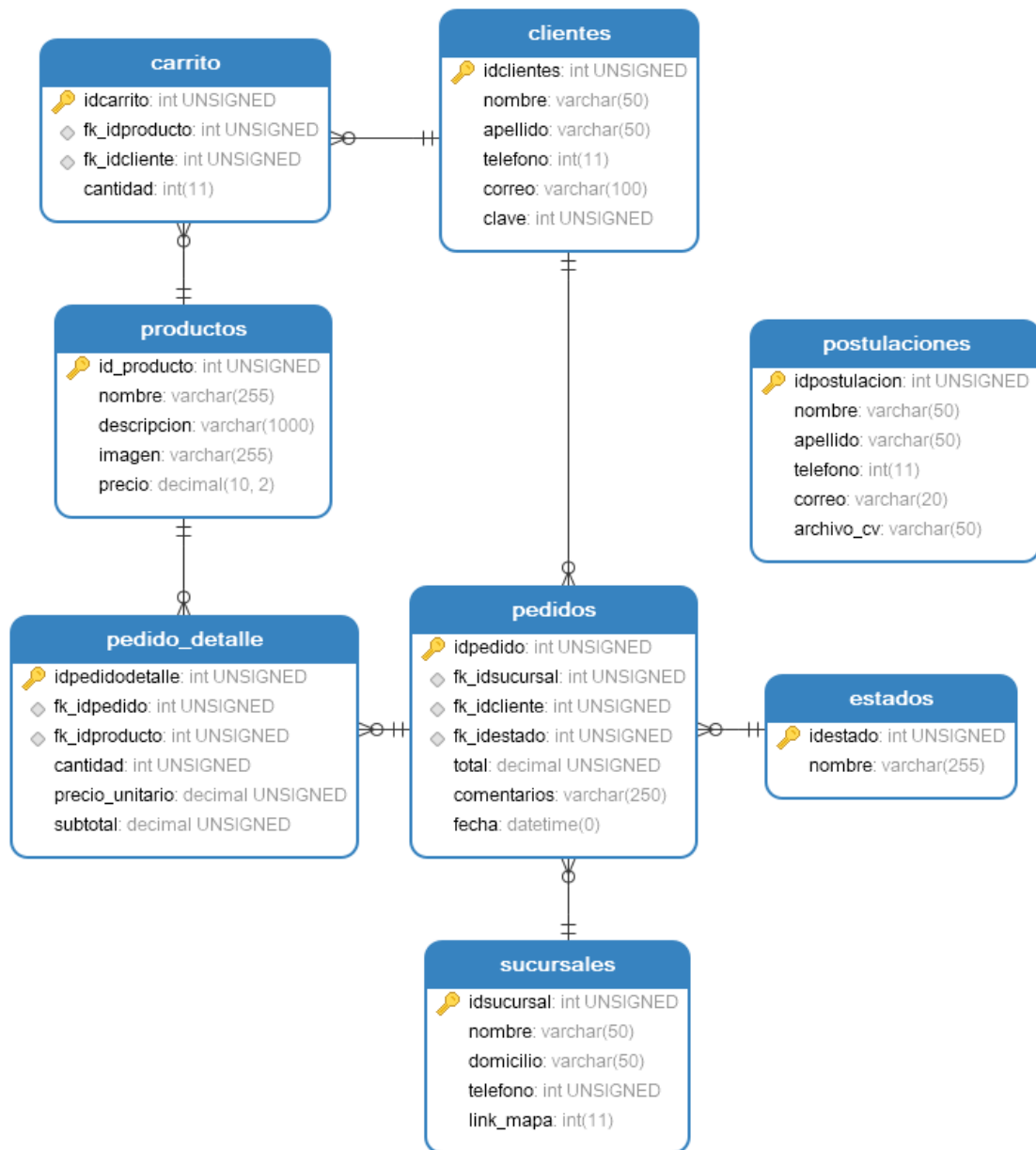
pedido\_detalle(idpedidodetalle, fk\_idpedido, fk\_idproducto, cantidad, precio\_unitario, subtotal) **tomas**

estado(idestadopedido, nombre) **maricel**

carrito(idcarrito, fk\_idproducto, fk\_idcliente, cantidad) **anderson**

postulaciones (idpostulacion, nombre, apellido, telefono, correo, archivo\_cv) **ezequiel**

usuarios (está resuelta)



## Base de datos

Alumnos que cursan en vivo (en la nube)	Alumnos libres (en su localhost)
Host: 167.114.86.211	Host: 127.0.0.1
Usuario: burgers_062021	Usuario: root
Clave: BURG.4579 (SOLO COPIAR, NO CORTAR)	Clave:
Puerto: 3310	Puerto: 3306
Nombre BBDD: burgers_062021	Nombre BBDD: sistema

### Repositorio en Github para alumnos en vivo

*El docente debe crear un repositorio llamado burger<ID\_del\_curso>, y agregar a los alumnos como colaboradores con permiso de escritura. El docente a partir de la plantilla del proyecto sube los nuevos archivos al repositorio. El alumno clonará el repositorio para comenzar a trabajar en el proyecto.*

Ejemplo:

git clone https://github.com/depcsuite/burger062021.git

### Repositorio en Github para alumnos libres (plantilla de solo lectura)

Descargar los archivos del siguiente repositorio y crear uno propio:

[https://github.com/depcsuite/plantilla\\_proyecto](https://github.com/depcsuite/plantilla_proyecto)

### Panel de administración

<http://127.0.0.1:8000/admin/logi>

Usuario: admin

Clave: 1234

### Usuario cliente de test

<http://127.0.0.1:8000/>

Usuario: sarah@gmail.com

Clave: 123456789

### Archivo de configuración .env

```
APP_NAME="Burgers SRL"
APP_ENV=local
APP_KEY=base64:xggbSM2IZ+yTAwTfsZT5iUKg033uMLT+aNCB1MVEVss
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://127.0.0.1:8000
APP_PATH="/Users/nelson/Desktop/Proyectos FS/burger062021"

DB_CONNECTION=mysql
DB_HOST=167.114.86.211
DB_PORT=3310
DB_DATABASE=burgers_062021
DB_USERNAME=burgers_062021
DB_PASSWORD=BURG.4579

CACHE_EXPIRE = -1
LOG_CHANNEL=none

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120
QUEUE_DRIVER=sync
```

	<p>             REDIS_HOST=127.0.0.1              REDIS_PASSWORD=null              REDIS_PORT=6379           </p> <p>             MAIL_DRIVER=smtp              MAIL_HOST=null              MAIL_PORT=25              MAIL_USERNAME=null              MAIL_PASSWORD=null              MAIL_ENCRYPTION=tls              MAIL_FROM_NAME=null              MAIL_FROM_ADDRESS=null              MAIL_NOREPLY=null           </p> <p>             PUSHER_APP_ID=              PUSHER_APP_KEY=              PUSHER_APP_SECRET=           </p>
--	---

## Menú administrador

Menú padre (Con icono)	Submenú	URL
Cientes (2)	Nuevo cliente (1) <b>Anderson</b>	/admin/cliente/nuevo
Cientes	Listado de clientes (2) <b>Anderson</b>	/admin/clientes
Productos (3)	Nuevo producto (1) <b>Marisel</b>	/admin/producto/nuevo
Productos	Listado de productos (2) <b>Marisel</b>	/admin/productos
Pedidos (4)	Nuevo pedido (1) <b>Tomás</b>	/admin/pedido/nuevo
Pedidos	Listado de pedidos (2) <b>Tomás</b>	/admin/pedidos
Sucursales (5)	Nueva sucursal (1) <b>Ezequiel</b>	/admin/sucursal/nuevo
Sucursales	Listado de sucursales (2) <b>Ezequiel</b>	/admin/sucursales
Postulaciones (6)	Nueva postulación (1) <b>Anderson</b>	/admin/postulacion/nuevo
Postulaciones	Listado de postulaciones (2) <b>Anderson</b>	/admin/postulaciones

- \*\* Los menús padre no tienen URL.
- \*\* Los menús padre tienen ícono y los sub menús no.
- \*\* Siempre marcar la casilla de sistemas, para asociar el menú al área de trabajo sistemas.

### Menú sitio institucional

Menú padre (Con ícono)	Submenú	URL
Mostrar productos en el admin	Tomás	

```
<form id="form1" method="POST" enctype="multipart/form-data">
```

```
public function guardar(Request $request)
{
    try {
        //Define la entidad servicio
        $titulo = "Guardar producto";
        $entidad = new Producto();
        $entidad->cargarDesdeRequest($request);

        if ($_FILES["imagen"]["error"] === UPLOAD_ERR_OK) { //Se adjunta imagen
            $nombre = date("Ymdhmsi");
            $archivo = $_FILES["imagen"]["tmp_name"];
            $extension = pathinfo($_FILES["imagen"]["name"], PATHINFO_EXTENSION);

            if($extension == "png" || $extension == "jpg" || $extension == "jpeg"){
                move_uploaded_file($archivo, env('APP_PATH') . "/public/files/$nombre.$extension");
            }
            //guardaelarchivo
            $producto->imagen = "$nombre.$extension";
        } else {
            $producto->imagen = "";
        }
    }
}
```



```
    }

    //validaciones
    if ($entidad->nombre == "") {
        $msg["ESTADO"] = MSG_ERROR;
        $msg["MSG"] = "Complete todos los datos";
    } else {
        if ($_POST["id"] > 0) {
            $productAnt = new Producto();
            $productAnt->obtenerPorId($entidad->idproducto);
            if($_FILES["archivo"]["error"] === UPLOAD_ERR_OK){
                //Eliminar imagen anterior
                @unlink(env('APP_PATH') . "/public/files/$productAnt->imagen");
            } else {
                $entidad->imagen = $productAnt->imagen;
            }

            //Es actualizacion
            $entidad->guardar();

            $msg["ESTADO"] = MSG_SUCCESS;
            $msg["MSG"] = OKINSERT;
        } else {
            //Es nuevo
            $entidad->insertar();

            $msg["ESTADO"] = MSG_SUCCESS;
            $msg["MSG"] = OKINSERT;
        }
        $_POST["id"] = $entidad->idproducto;
        return view('producto.producto-listar', compact('titulo', 'msg'));
    }
} catch (Exception $e) {
    $msg["ESTADO"] = MSG_ERROR;
    $msg["MSG"] = ERRORINSERT;
}

$id = $entidad->idproducto;
$producto = new Producto();
$producto->obtenerPorId($id);

return view('producto.producto-nuevo', compact('msg', 'producto', 'titulo')) . '?id=' .
$producto->idproducto;
}
```

## Envío de correo electrónico

### //Cabeceras

```
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
```

```
//Dentro del método de envío
```

```
//Instancia y configuraciones de PHPMailer
```

```
$mail = new PHPMailer(true);
$mail->SMTPDebug = 0;
$mail->isSMTP();
$mail->Host = env('MAIL_HOST');
$mail->SMTPAuth = true;
$mail->Username = env('MAIL_USERNAME');
$mail->Password = env('MAIL_PASSWORD');
$mail->SMTPSecure = env('MAIL_ENCRYPTION');
$mail->Port = env('MAIL_PORT');
```

```
//Destinatarios
```

```
$mail->setFrom(env('MAIL_FROM_ADDRESS'), env('MAIL_FROM_NAME')); //Dirección desde
$mail->addAddress($email); //Dirección para
$mail->addReplyTo($replyTo); //Dirección de reply no-reply
$mail->addBCC($copiaOculta); //Dirección de CCO
```

```
//Contenido del mail
```

```
$mail->isHTML(true);
$mail->Subject = $subject;
$mail->Body = $body;
//$mail->send();
```

## Agregar en formulario-cliente.php

```
if(isset($_GET["do"]) && $_GET["do"] == "cargarGrilla"){
    $idCliente = $_GET['idCliente'];
    $request = $_REQUEST;

    $entidad = new Domicilio();
    $aDomicilio = $entidad->obtenerFiltrado($idCliente);

    $data = array();

    $inicio = $request['start'];
    $registros_por_pagina = $request['length'];

    if (count($aDomicilio) > 0)
```

```
$cont=0;
for ($i=$inicio; $i < count($aDomicilio) && $cont < $registros_por_pagina; $i++) {
    $row = array();
    $row[] = $aDomicilio[$i]->tipo;
    $row[] = $aDomicilio[$i]->provincia;
    $row[] = $aDomicilio[$i]->localidad;
    $row[] = $aDomicilio[$i]->domicilio;
    $cont++;
    $data[] = $row;
}

$json_data = array(
    "draw" => intval($request['draw']),
    "recordsTotal" => count($aDomicilio), //cantidad total de registros sin paginar
    "recordsFiltered" => count($aDomicilio), //cantidad total de registros en la paginacion
    "data" => $data
);
echo json_encode($json_data);
exit;
}
```

## Agregar en entidad domicilio

```
public function obtenerFiltrado($idCliente){
    $mysqli = new mysqli(Config::BBDD_HOST, Config::BBDD_USUARIO, Config::BBDD_CLAVE,
Config::BBDD_NOMBRE);
    $request = $_REQUEST;
    $columns = array(
        0 => 'B.nombre',
        1 => 'D.nombre',
        2 => 'C.nombre',
        3 => 'A.domicilio'
    );
    $sql = "SELECT
        A.iddomicilio,
        A.fk_tipo,
        B.nombre AS tipo,
        A.fk_idlocalidad,
        C.nombre AS localidad,
        D.idprovincia,
        D.nombre AS provincia,
        A.domicilio
    FROM domicilios A
    INNER JOIN tipo_domicilios B ON A.fk_tipo = B.idtipo
    INNER JOIN localidades C ON C.idlocalidad = A.fk_idlocalidad
    INNER JOIN provincias D ON D.idprovincia = C.fk_idprovincia
    WHERE 1=1 AND A.fk_idcliente = $idCliente
    ";

    //Realiza el filtrado
    if (!empty($request['search']['value'])) {
        $sql.=" AND ( B.nombre LIKE '%" . $request['search']['value'] . "%' ";
        $sql.=" OR D.nombre LIKE '%" . $request['search']['value'] . "%' ";
        $sql.=" OR C.nombre LIKE '%" . $request['search']['value'] . "%' ";
        $sql.=" OR A.domicilio LIKE '%" . $request['search']['value'] . "%' )";
    }
    $sql.=" ORDER BY " . $columns[$request['order'][0]['column']] . " " . $request['order'][0]['dir'];

    $resultado = $mysqli->query($sql);
    $lstRetorno = array();
    while ($fila = $resultado->fetch_assoc()) {
        $domicilio = new Domicilio();
        $domicilio->iddomicilio = $fila["iddomicilio"];
        $domicilio->fk_tipo = $fila["fk_tipo"];
        $domicilio->tipo = $fila["tipo"];
        $domicilio->fk_idlocalidad = $fila["fk_idlocalidad"];
```

```
$domicilio->localidad = $fila["localidad"];
$domicilio->idprovincia = $fila["idprovincia"];
$domicilio->provincia = $fila["provincia"];
$domicilio->domicilio = $fila["domicilio"];
$IstRetorno[] = $domicilio;
}
return $IstRetorno;
}
```