|

SISTEMA

**Índice**

**Contenido**

[**INTRODUCCIÓN**](https://docs.google.com/document/d/16aJ9OI5VTBFLOyg0dxn0-tGmWnSzYqju4YgNNRvyJaY/edit#heading=h.vzeeeznv1vkj)[**2**](https://docs.google.com/document/d/16aJ9OI5VTBFLOyg0dxn0-tGmWnSzYqju4YgNNRvyJaY/edit#heading=h.vzeeeznv1vkj)

[**OBJETIVOS**](https://docs.google.com/document/d/16aJ9OI5VTBFLOyg0dxn0-tGmWnSzYqju4YgNNRvyJaY/edit#heading=h.1fob9te)[**2**](https://docs.google.com/document/d/16aJ9OI5VTBFLOyg0dxn0-tGmWnSzYqju4YgNNRvyJaY/edit#heading=h.1fob9te)

[**DESARROLLO**](https://docs.google.com/document/d/16aJ9OI5VTBFLOyg0dxn0-tGmWnSzYqju4YgNNRvyJaY/edit#heading=h.hhemfxz2hvww)[**2**](https://docs.google.com/document/d/16aJ9OI5VTBFLOyg0dxn0-tGmWnSzYqju4YgNNRvyJaY/edit#heading=h.hhemfxz2hvww)

# 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”, “Entregado” y “Cancelado”

# REQUISITOS

**Requisitos sitio web comercial**

*Enumerar y detallar los distintos requisitos sobre lo que debe realizar el sitio institucional.*

*El sitio web debe tener un* ***menú*** *con los datos: Inicio, Takeaway,Nosotros,Contacto,Mi cuenta (solo si está logueado)/Iniciar sesión y botón carrito.*

***La pantalla takeaway*** *debe mostrar un listado de productos con los datos: título, descripción, precio,imagen, input de cantidad y botón para agregar al carrito.*

***La pantalla Nosotros*** *debe mostrar: banner principal con título, subtítulo, descripción de la empresa,.*

*slider de bootstrap con frases rotativas, Sección trabaja con nosotros donde tendrá formulario que solicita email, nombre, apellido, whatsapp, input para adjuntar Cv(pdf, docx, doc) y botón enviar.*

*pantalla con mensaje “Gracias, pronto estarás recibiendo noticias nuestras”*

**La pantalla Carrito:**Un usuario logueado, puede agregar productos al carrito, caso contrario indicar que tiene que iniciar sesión. Debe listar todos los productos que el usuario haya agregado, podrá eliminar productos del carrito, seleccionar la sucursal de retiro, finalizar pedido para confirmar. Agregar desplegable con modalidad de pago “Efectivo” o “Mercado Pago” el botón continuar pedido regresa a la página de listado de productos (takeaway)

**La pantalla Contacto:** Debe mostrar un formulario de contacto con nombre, correo electrónico, número de teléfono y campo de texto para comentarios.

**La pantalla Mi Cuenta**: debe mostrar y poder editar los datos del usuario: nombre, apellido, celular, whatsapp, email, y una tabla con los pedidos activos.

Desde una pantalla aparte, se debe permitir cambiar la contraseña,ingresando la contraseña anterior y nueva siempre y cuando sea valida.

**La pantalla de Registro:** debe permitir crear una cuenta al usuario con los daatos: nombres, apellido, celular, email y clave.

**La Home**:Debe mostrar banner de hamburguesas, mostrar los domicilios de todas las sucursales desde la bbdd con número de teléfono, dirección y boton con link de google maps(al menos 4 sucursales)

**Requisitos para el panel administrativo**

*Enumerar y detallar los distintos requisitos sobre lo que debe realizar el panel administrativo.(datos de tablas)*

*El sistema debe permitir Agregar, Actualizar, Eliminar* ***productos*** *con los campos: título, descripción, importe, cantidad, imagen.*

*El sistema debe permitir Agregar, Actualizar, Eliminar* ***categorías*** *con los campos: nombre..*

*El sistema debe permitir Agregar, Actualizar, Eliminar* ***sucursales*** *con los campos: nombre, dirección, teléfono, mapa.*

*El sistema debe permitir Agregar, Actualizar, Eliminar* ***clientes*** *con los campos: nombre, email, dni, celular*

*El sistema debe permitir Agregar, Actualizar, Eliminar* ***postulaciones*** *con los campos: nombre, apellido, whatsapp, correo,link*

*El sistema debe permitir actualizar el* ***estado de un pedido*** *a: Pendiente, En preparación, Terminado, Cancelado.*

*El sistema debe permitir listar los* ***pedidos*** *en una grilla con los datos: ver detalle del pedido, número de pedido, sucursal, cliente, celular, fecha, total, método de pago;*

*El sistema debe permitir listar los* ***productos*** *en una grilla con los datos: nombre, categoría, precio, pesos.*

*El sistema debe permitir listar los* ***clientes*** *en una grilla con los datos: nombre, apellido, email, dni. celular.*

*El sistema debe permitir listar las* ***sucursales*** *en una grilla con los datos: dirección, teléfono, mapa, nombre y horario.*

*El sistema debe permitir listar los* ***postulantes*** *en una grilla con los datos: botón ver postulación, nombre, apellido, whatsapp, correo, y link para descargar el CV.*

**Páginas de la competencia**

| **URL** | **Alumno** |
| --- | --- |
| https://www.mcdonalds.com.ar/ |  |
| https://www.mostazaweb.com.ar/ |  |
| https://www.labirrabar.com/ |  |

**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://themewagon.com/theme-framework/bootstrap-5/)

[*https://bootstrapmade.com/*](https://bootstrapmade.com/)

| **URL** | **Alumno** |
| --- | --- |
|  |  |
|  |  |

# DESARROLLO

**Tablas del sistema**

**productos** ( idproducto, nombre, precio,descripción, imagen, fk\_idcategoria, cantidad)

**pedidosproductos** ( idpedidoproducto, nombre, fk\_idpedido, fk\_idproducto, cantidad, precio\_unitario, total)

**clientes** (idcliente,nombre, apellido, correo, dni,celular, clave)

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

**sucursales**(idsucursal, nombre, dirección,teléfono, linkmapa)

**estados**(idestado, nombre)

**categorias**(idcategoria,nombre)

**carrito\_productos**(idcarrito\_prducto, fk\_idproducto, fk\_idcarrito,cantidad)

**postulaciones**(idpostulacion, nombre, apellido,celular, correo,curriculum)

Datos maestros: Estado (En preparación,Pendiente,Cancelado, Finalizado)

**Base de datos**

| Alumnos que cursan en vivo (en la nube) | Alumnos libres (en su localhost) |
| --- | --- |
| Host: 167.114.86.211  Usuario: burgers\_062021  Clave: BURG.4579 (SOLO COPIAR, NO CORTAR)  Puerto: 3310  Nombre BBDD: burgers\_062021 | Host: 127.0.0.1  Usuario: root  Clave:  Puerto: 3306  Nombre BBDD: sistema |

**Repositorio**

actualizar .gitignore y vaciar logs

git config --global user.email "nelson.tarche@mail.com"

git config --global user.name "ntarche"

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

**Panel de administración**

[http://127.0.0.1:8000/admin/logi](http://127.0.0.1:8000/admin/login)

Usuario: admin

Clave: 1234

**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/ProyectosFS/burger022022'  APP\_PATH='C:\Users\USUARIO\Desktop\burger022022'  DB\_CONNECTION=mysql  DB\_HOST=167.114.86.211  DB\_PORT=3310  DB\_DATABASE=burgers\_042021  DB\_USERNAME=burgers\_042021  DB\_PASSWORD=BR8K.8496  CACHE\_EXPIRE = -1  LOG\_CHANNEL=none  BROADCAST\_DRIVER=log  CACHE\_DRIVER=file  SESSION\_DRIVER=file  SESSION\_LIFETIME=120  QUEUE\_DRIVER=sync  REDIS\_HOST=127.0.0.1  REDIS\_PASSWORD=null  REDIS\_PORT=6379  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  PUSHER\_APP\_ID=  PUSHER\_APP\_KEY=  PUSHER\_APP\_SECRET= |
| --- | --- |

**Menú administrador**

| **Menú padre**  **(Con icono)** | **Submenú** | **URL** |
| --- | --- | --- |
| Productos (1)/posición/ | Nuevo producto(1)  Listado de productos(2) | /admin/productos  /admin/producto/nuevo |
| Clientes (2) | Nuevo cliente(1)  Listado de clientes(2) | /admin/clientes  /admin/cliente/nuevo |
| Pedidos (3) | Nuevo pedido(1)  Listado de pedidos(2) | /admin/pedidos  /admin/pedido/nuevo |
| Postulaciones(4) | Nueva postulación(1)  Listado de postulaciones(2) | /admin/postulaciones  /admin/postulación/nuevo |
| Sucursales (5) | Nueva sucursal(1)  Listado de sucursales(2) | /admin/sucursales  /admin/sucursal/nuevo |
| Categorías (6) | Nueva categoría(1)  Listado de categorías(2) | /admin/categorías  /admin/categoría/nueva |
| Proveedores(7) | Nuevo proveedor(1)  Listado de proveedores(2) | /admin/proveedores  /admin/proveedor/nuevo |

**Menú sitio institucional**

| **Menú padre**  **(Con icono)** | **Submenú** | **URL** |
| --- | --- | --- |
| inicio |  | / |
| Takeaway |  | /takeaway |
| Nosotros |  | /nosotros |
|  | Gracias por postularte | /postulacion-gracias |
| Carrito |  | /carrito |
| Mi cuenta |  | /mi-cuenta |
|  | Cambiar clave | /cambiar-clave |
| Contacto |  | /contacto |
|  | Gracias por contactarse | /contacto-gracias |
| Login |  | /login |
| Registrarse |  | /registrarse |
| Olvidé la clave |  | /recuperar-clave |

<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["archivo"]["error"] === UPLOAD\_ERR\_OK) { //Se adjunta imagen

$extension = pathinfo($\_FILES["archivo"]["name"], PATHINFO\_EXTENSION);

$nombre = date("Ymdhmsi") . ".$extension";

$archivo = $\_FILES["archivo"]["tmp\_name"];

move\_uploaded\_file($archivo, env('APP\_PATH') . "/public/files/$nombre"); //guardaelarchivo

$entidad->imagen = $nombre;

}

//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;

}

**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"];

$lstRetorno[] = $domicilio;

}

return $lstRetorno;

}

Nicolas: Página gracias postulacion

Alejandro: fontawesome, En Mi cuenta mostrar los datos del cliente logueado y que pueda cambiarlos

Maximo: Eliminar del carrito

Matías: Corregir importe del pedido en el admin, mostrar productos del pedido detalle, y vaciar carrito

Nicolás: Promociones

Lucas: Sponsor