|

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 institucional**

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

**El sitio web** debe mostrar logotipo del comercio menú, pie de página, página de inicio.

El sitio web debe tener las siguientes opciones de **menú**:

* Inicio,
* Takeaway,
* Nosotros,
* Contacto,
* Mi cuenta
* Carrito de compra (botón o imagen).

**La página de inicio** debe contener banner de hamburguesa, mostrar domicilios de las sucursales traídas desde la BBDD con número de teléfono, dirección, y botón de link a google maps (al menos 4 sucursales).

**La página Takeaway** debe mostrar el listado de productos. Por producto deberá contener:

* Título
* Descripción
* Precio
* Input de cantidad
* Y botón de agregar al carrito

Este listado de productos deberá traerlo desde la BBDD y agregar al carrito tiene que guardarse en la misma.

**La página nosotros** contendrá:

* Banner principal de la sección con un título
* Subtítulo: descripción de la empresa y un slider de bootstrap con frases rotativas
* Seccion trabaja con nosotros: Formulario que solicita email, nombre, apellido, WhatsApp, input para adjuntar CV(pdf, docx, doc) y botón enviar. Una vez completado deberá aparecer una pantalla un mensaje: “Gracias, pronto estarás recibiendo noticias nuestras”. Tiene que registrarse este contacto en la base de datos como una postulación.

**La página contacto** debe tener un formulario con los campos nombre, apellido, celular, email, mensaje

**La página mi cuenta** deberá mostrar y poder editar los datos del usuario tales como nombre, apellido, celular, WhatsApp, email y una tabla de pedidos activos.

(DAR DE ALTA EN LA BBDD UNA TABLA DE SUCURSALES PREDETERMINADAS, HACER UN ABM DE SUCURSALES.)

La página de carrito de compras deberá traer de la BBDD que tenga un usuario. Del carrito cada usuario podrá:

* Eliminar un producto
* Seleccionar sucursal de retiro
* Finalizar el pedido para confirmar. (agregar desplegable con forma de pago en sucursal o MercadoPago.)
* Botón continuar que regresa al listado de productos (Takeaway)

**Requisitos para el panel administrativo**

*Enumerar y detallar los distintos requisitos sobre lo que debe realizar el panel administrativo.*

El sistema deberá permitir el alta, baja, consulta y modificación de:

* **Productos:** contener una tabla de jquery datatable con los datos nombre, categoría, precio.

Formulario: título, descripcion, importe, para dar de alta o editar un nuevo producto.

* **Clientes:** deberá contener una tabla de jquery datatable con los datos nombre, apellido, email, dni, celular.

Formulario (nuevo/editar)

* **Pedidos:** deberá contener una tabla de jquery datatable con los datos botón de vista n° de pedido, sucursal, cliente, celular, fecha, total.

El botón de vista deberá poder ver el listado de productos solicitados e importe a cobrar al cliente. Cambiar desde un select el estado del pedido de pendiente a entregado o cancelado.

* **Postulacione:** deberá contener una tabla de jquery datatable las postulaciones realizadas desde la web con las columnas: boton “ver postulación”, nombre, apellido, whatapps, correo y link para descargar CV.
* **Sucursales:** deberá contener una tabla de jquery datatable nombr, teléfono, dirección,
* **Páginas de la competencia**

|  |  |
| --- | --- |
| **URL** | **Alumno** |
| https://www.perez-h.com/newsite/ | Juan Mc Kenna |
| https://hormiganegra.com.ar/ | Juan Mc Kenna |
| https://www.cervelar.com.ar/ | Juan Mc Kenna |

**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**

**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/Proyectos FS/broker-seguros"  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** |
| Inicio (1) |  |  |
| Clientes (2) | Listado de clientes (1)  Nuevo Cliente (2) | /admin/clientes  /admin/clientes/nuevo |
| Productos (3) | Listado de productos (1)  Nuevo producto (2) | /admin/productos  /admin/productos/nuevo |
| Pedidos (4) | historial de pedidos (1)  Nuevo pedido (2) | /admin/pedidos  /admin/pedidos/nuevo |
| Categorías (5) | Listado de categorías (1)  Nueva categoría (2) | /admin/categorias  /admin/categorias/nuevo |
| Sucursales (6) | Listado de sucursales (1)  Nueva sucursal (2) | /admin/sucursales  /admin/sucursales/nuevo |
| Estados (7) | Listado de estados (1)  Nuevo estado (2) | /admin/estados  /admin/estados/nuevo |
| Postulaciones (8) | Listado de postulaciones (1)  Nueva postulacion (2) | /admin/postulaciones  /admin/postulaciones/nuevo |

**Menú sitio institucional**

|  |  |  |
| --- | --- | --- |
| **Menú padre**  **(Con icono)** | **Submenú** | **URL** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

<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

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

$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