

Universidad Autónoma de San Luís Potosí

Manual de programador

Alumno: Mendoza Rodríguez Fernando

Carrera: Ing. Sistemas inteligentes

Materia: Aplicaciones Web Interactivas

Profesora: Ing. Estrada Velázquez Francisco Everardo

Fecha: Domingo 10 de diciembre de 2021

Migraciones

Tabla de usuarios

```
class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id("idUserario");

            $table->string('usuario')->unique();
            $table->string('email')->unique()->nullable();
            $table->string("name");
            $table->string("apellido");
            $table->boolean("admin");

            $table->string('password');
            $table->timestamps();
        });
    }
    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

Tabla de ventas

```
class CreatePasswordResetsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
}
```

```

public function up()
{
    Schema::create('password_resets', function (Blueprint $table) {
        $table->string('email')->index();
        $table->string('token');
        $table->timestamp('created_at')->nullable();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('password_resets');
}
}

```

Tabla de productos

```

class CreateProductosTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('productos', function (Blueprint $table) {
            $table->id("idProducto");
            $table->string("codigoProd")->unique()->nullable();
            $table->string("nombre")->nullable();
            $table->float("precio",10,4);
            $table->integer("stock")->nullable();
            $table->boolean("eliminado");
            $table->string("direccionImg")->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
}

```

```

    *
    * @return void
    */
    public function down()
    {
        Schema::dropIfExists('productos');
    }
}

```

Tabla de detalle de venta

```

class CreateDetalleVentasTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('detalle_ventas', function (Blueprint $table) {
            $table->id("idDetalle");
            $table->unsignedBigInteger("idVenta");
            $table->unsignedBigInteger("idProducto");
            $table->integer("cantProd");
            $table->timestamps();
            $table->foreign('idVenta')->references('idVenta')->on('ventas');
            $table->foreign('idProducto')->references('idProducto')->
on('productos');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('detalle_ventas');
    }
}

```

Modelo

Detalle de venta

```
class DetalleVenta extends Model
{
    //
    protected $primaryKey = "idDetalle";
}
```

Producto

```
class Producto extends Model
{
    //protected $primaryKey = "CodigoProd";
    protected $primaryKey = "idProducto";
    //protected $keyType = "string";
}
```

Usuario

```
class User extends Authenticatable
{
    use Notifiable;
    protected $primaryKey = "idUsuario";
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'apellido', 'usuario', 'password', 'admin', 'email'
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password',
    ];
    /*
     * protected $hidden = [
     *     'password', 'remember_token',
     */
```

```

]; */

/**
 * The attributes that should be cast to native types.
 *
 * @var array
 */
/*
protected $casts = [
    'email_verified_at' => 'datetime',
]; */
}

```

Controladores

```

class Controller extends BaseController
{
    //use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
    public function welcome()
    {
        //return redirect("/alta");
        return view("welcome");
    }
}

```

Detalle de venta

```

class DetalleVentaController extends Controller
{
    //
    public function registraDetalle($idVenta, $idProducto,$cantidad)
    {
        $detalle = new DetalleVenta();
        $detalle->idVenta=$idVenta;
        $detalle->idProducto=$idProducto;
        $detalle->cantProd=$cantidad;
        $detalle->save();
    }
}

```

Producto

```
class ProductoController extends Controller
{
    //
    public function alta()
    {
        if(Auth::check() && Auth::user()->admin==1)
        {
            return view("altaDeProducto");
        }
        return redirect("/home");
    }

    public function guardaProducto(Request $request)
    {
        //dd($request);
        if(isset($request->idProducto))
        {
            $producto = Producto::find($request->idProducto);
        }
        else
        {
            $producto = new Producto();
        }

        $producto->codigoProd = $request->codigoProd;
        $producto->nombre = $request->nombre;
        $producto->precio = $request->precio;

        if(is_null($request->stock))
        {
            $producto->stock = 0;
        }
        else
        {
            $producto->stock = $request->stock;
        }
        if(!is_null($request->nombreImg))
        {
            $producto->direccionImg = $request->file("imagen")-
>store($request->nombreImg); //imagen es el name en el input
        }
        else
        {
            if(!isset($request->idProducto))
                $producto->direccionImg = NULL;
        }
    }
}
```

```

        $producto->eliminado = False;

        $producto->save();
        return redirect()->back();
    }

    public function buscaProducto(Request $request)
    {
        if($request->input("cadena")!=null)
        {
            $cadena=$request->input("cadena");
            //dd($cadena);
            $cadena = "%".$cadena."%";
            $productos=Producto::where("eliminado", false)-
>where("codigoProd","like",$cadena)->orWhere("nombre",'like',$cadena)-
>get();
            return view("busqueda")->with("productos",$productos);
        }
        else
            return redirect()->back();
    }

    public function muestraProducto($idProducto=null)
    {
        if(!is_null($idProducto))
        {
            $producto = Producto::find($idProducto);
            //dd($producto);
            if($producto->eliminado==true)
                return redirect("/");
            return view("visualizaProducto")->with("producto",$producto);
        }
        return redirect("/");
    }

    public function eliminaProducto(Request $request)
    {
        #Verificar que el usuario sea administrador,
        if(Auth::check() && Auth::user()->admin==1)
        {

```



```

        $idProducto=$request->idProducto;
        $producto=Producto::find($idProducto);
        //dd($producto);
        $producto->Eliminado=True;
        //dd($producto);
        $producto->save();
    }
    //return redirect()->back();
    return redirect("/");
}

public function cajaAgregaProd(Request $request)
{
    if(isset($request->codigoProd))
    {
        $producto = Producto::where("codigoProd",$request->codigoProd)-
>where("eliminado",0)->first();
        return response(json_encode($producto),200)->header("Content-
type","text/plain");
    }
}

public function descuentaproducto(int $id, int $cantidad)
{
    $producto=Producto::where("idProducto",$id)-
>decrement("stock",$cantidad);
}
}

```

Venta

```

class VentaController extends Controller
{
    //

    public function vendeProducto($idProducto=null)
    {
        //dd($idProducto);
    }
}

```

```

        if(!is_null($idProducto))
        {
            return view("caja")->with("idProducto",$idProducto);
        }
        return redirect("/");
    }

    public function muestraCaja()
    {
        return view("caja");
    }

    public function realizaVenta(Request $request)
    {
        //$listId=[];
        //$listCantidades=[];
        //Bandera para creación de venta
        $ventaBandera=false;
        $venta=null;

        for ($i=1; ; $i++) {
            $numProducto=strval($i);

            if(!$ventaBandera)
            {
                $venta = new Venta();
                $venta->idUsuario = Auth::user()->idUsuario;
                $fecha = new DateTime();
                $venta->fechaCompra = $fecha->format('Y-m-d H:i:s');
                $venta->save();
                $ventaBandera=true;
            }

            if($request->input($numProducto)!=null)
            {
                $idProd= intval($request->input($numProducto));
                $cantidad=intval($request->input("cantidad".$numProducto));
                //dd($cantidad);
                app('App\Http\Controllers\ProductoController')->descuentaProducto($idProd,$cantidad);
            }
        }
    }

```

```

        app('App\Http\Controllers\DetalleVentaController')->
        >registraDetalle($venta->idVenta,$idProd,$cantidad);
    }
    else
        break;
    }
    return redirect("/caja");
}

public function muestraReporte()
{
    //Primero tengo que obtener los productos mas vendidos
    $Productos = DB::table('productos')
    -
    >join('detalle_ventas','detalle_ventas.idProducto','=','productos.idProducto')
    -
    >select("productos.idProducto","productos.codigoProd","productos.nombre",DB::
    :raw("sum(cantProd) as sum"))
    ->groupBy("idProducto")->orderBy(DB::raw("sum(cantProd)"),"desc")-
    >take(10)->get()->toArray();

    //Días ordenados por cual tiene mas ventas
    $dias = DB::table('ventas')
    ->select(DB::raw("DATE_FORMAT(fechaCompra,'%W') as
    dia"),DB::raw("count(*) as sum"))
    ->groupBy(DB::raw("DATE_FORMAT(fechaCompra,'%W')"))-
    >orderBy(DB::raw("count(*)"),"desc")->get()->toArray();
    //dd($dias);
    //dd($Productos);
    //dd(gettype($Productos));

    $data=[$Productos,$dias];
    //return view("reporte")->with('data',$data);
    $pdf = PDF::loadView('reporte',compact('Productos','dias') );
    return $pdf->download('invoice.pdf');
}
}

```

Rutas

```
Route::get("/alta", "ProductoController@alta")->middleware("auth");
Route::post("/guardaProducto", "ProductoController@guardaProducto")->middleware("auth");
//Read, Update, Delete
Route::get("/producto/{idProducto}", "ProductoController@muestraProducto")->middleware("auth");
Route::post("/eliminar", "ProductoController@eliminaProducto")->middleware("auth");
Route::get("/buscar", "ProductoController@buscaProducto")->middleware("auth");

Route::get("/vender/{idProducto}", "VentaController@vendeProducto")->middleware("auth");
Route::get("/caja", "VentaController@muestraCaja")->middleware("auth");

Route::post("/caja/agregaProd", "ProductoController@cajaAgregaProd")->middleware("auth");
Route::post("/caja/realizaVenta", "VentaController@realizaVenta")->middleware("auth");

Route::post("/retornaProdCaja/{idProdCu}");

Route::get("/reportes", "VentaController@muestraReporte");

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');
```

Vistas

Archivo	Función
altaDeProducto.blade.php	Esta es la vista para dar de alta un producto
busqueda.blade.php	Esta es la vista para buscar productos
caja.blade.php	Esta es la vista para la caja registradora
reporte.blade.php	Esta es la vista del reporte
visualizaProducto.blade.php	Esta es la vista del producto
welcome.blade.php	Esta es la vista de entrada
register.blade.php	Esta es la vista para registrar un usuario