SENATI

SERVICIO NACIONAL DE ADIESTRAMIENTO EN TRABAJO INDUSTRIAL

FORMACIÓN PROFESIONAL DUAL

INFORME DE PRÁCTICA

CÓDIGO N° 89001677

DIRECCIÓN ZONAL

FORMACIÓN PROFESIONAL DUAL

CFP/UCP/ESCUELA: SENATI

ESTUDIANTE: Gianfranco Huillca Jaimez

ID: 1440238 BLOQUE: **NRC_32607**

CARRERA: informática y desarrollo de aplicaciones web

INSTRUCTOR: GIANCARLOS BARBOZA NIETO

SEMESTRE: 6 semestre

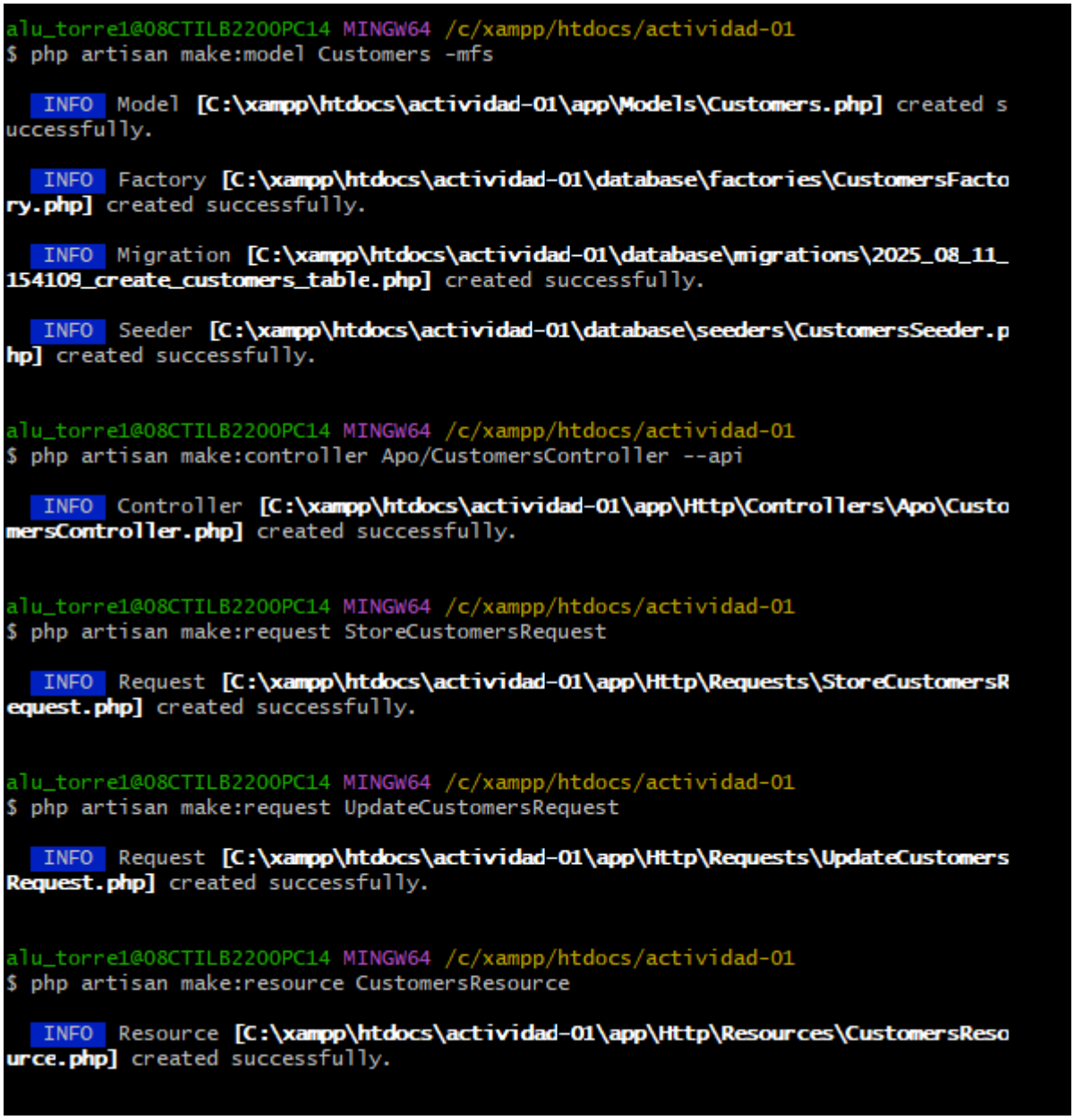
1) **El código que usaron en Git Bash para crear las tablas Clientes, Pedidos y Categorías.**

Creacion de las tablas en el git dash para eso usamos estos códigos

Aquí vamos a crear a make el model, Controller, Request, resource de la parte de los Productos(products).

```
alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:model Products -mfs

  INFO  Model [C:\xampp\htdocs\actividad-01\app\Models\Products.php] created su
ccessfully.

  INFO  Factory [C:\xampp\htdocs\actividad-01\database\factories\ProductsFactor
y.php] created successfully.

  INFO  Migration [C:\xampp\htdocs\actividad-01\database\migrations\2025_08_11_
160803_create_products_table.php] created successfully.

  INFO  Seeder [C:\xampp\htdocs\actividad-01\database\seeders\ProductsSeeder.ph
p] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:Controller Apo/ProductsController --api

  INFO  Controller [C:\xampp\htdocs\actividad-01\app\Http\Controllers\Apo\Produ
ctsController.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisanmake:request StoreProductsRequest
Could not open input file: artisanmake:request

alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:request StoreProductsRequest

  INFO  Request [C:\xampp\htdocs\actividad-01\app\Http\Requests\StoreProductsRe
quest.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:request UpdateProductsRequest

  INFO  Request [C:\xampp\htdocs\actividad-01\app\Http\Requests\UpdateProductsR
equest.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:resource ProductsResource

  INFO  Resource [C:\xampp\htdocs\actividad-01\app\Http\Resources\ProductsResou
rce.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan migrate

  INFO  Running migrations.

  2025_08_11_160803_create_products_table ..................... 9.32ms DONE
```

Ahora vamos con la parte De Customers(Clientes)



Ahora la parte del las categories (Categorias)

```
alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:model Categories -mfs

  INFO  Model [C:\xampp\htdocs\actividad-01\app\Models\Categories.php] created
successfully.

  INFO  Factory [C:\xampp\htdocs\actividad-01\database\factories\CategoriesFact
ory.php] created successfully.

  INFO  Migration [C:\xampp\htdocs\actividad-01\database\migrations\2025_08_11_
154414_create_categories_table.php] created successfully.

  INFO  Seeder [C:\xampp\htdocs\actividad-01\database\seeders\CategoriesSeeder.
php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:controller Apo/CategoriesController --api

  INFO  Controller [C:\xampp\htdocs\actividad-01\app\Http\Controllers\Apo\Categ
oriesController.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:request StoreCategoriesRequest

  INFO  Request [C:\xampp\htdocs\actividad-01\app\Http\Requests\StoreCategories
Request.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:request UpdateCategoriesRequest

  INFO  Request [C:\xampp\htdocs\actividad-01\app\Http\Requests\UpdateCategorie
sRequest.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:resource CategoriesResource

  INFO  Resource [C:\xampp\htdocs\actividad-01\app\Http\Resources\CategoriesRes
ource.php] created successfully.
```

Ahora la parte de Pedidos (Orders)

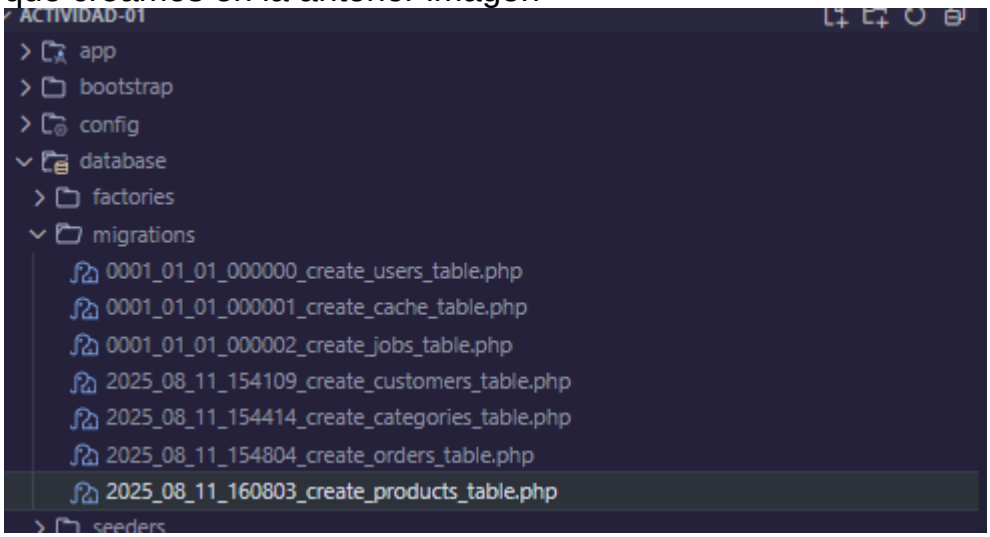
```
alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:model Orders -mfs

  INFO  Model [C:\xampp\htdocs\actividad-01\app\Models\Orders.php] created succ
essfully.

  INFO  Factory [C:\xampp\htdocs\actividad-01\database\factories\OrdersFactory.
php] created successfully.

  INFO  Migration [C:\xampp\htdocs\actividad-01\database\migrations\2025_08_11_
154804_create_orders_table.php] created successfully.

  INFO  Seeder [C:\xampp\htdocs\actividad-01\database\seeders\OrdersSeeder.php]
 created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:controller Apo/OrdersController --api

  INFO  Controller [C:\xampp\htdocs\actividad-01\app\Http\Controllers\Apo\Order
sController.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:request StoreOrdersRequest

  INFO  Request [C:\xampp\htdocs\actividad-01\app\Http\Requests\StoreOrdersRequ
est.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:request UpdateOrdersRequest

  INFO  Request [C:\xampp\htdocs\actividad-01\app\Http\Requests\UpdateOrdersReq
uest.php] created successfully.


alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan make:resource OrdersResource

  INFO  Resource [C:\xampp\htdocs\actividad-01\app\Http\Resources\OrdersResourc
e.php] created successfully.
```

Y aquí migramos Las categorías, Pedidos y Clientes

```
alu_torre1@O8CTILB2200PC14 MINGW64 /c/xampp/htdocs/actividad-01
$ php artisan migrate:rollback

  INFO  Rolling back migrations.

  2025_08_11_154804_create_orders_table ..................... 10.79ms DONE
  2025_08_11_154414_create_categories_table ..................... 5.44ms DONE
  2025_08_11_154109_create_customers_table ..................... 4.59ms DONE
```

2)El código que usaron en Visual Studio Code para crear los diferentes campos de cada
tabla.

Aquis mostraremos los códigos de las tablas del visual estudio que esta son los que creamos en la anterior imagen



Ahora los códigos de las migrations de Producto, orders, categories y customers

```php
Schema::create('products', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->text('descripcion');
    $table->string('sku')->unique();
    $table->integer('stock')->default(0);
    $table->decimal('price', 10, 2);
    $table->boolean('active')->default(false);
    $table->softDeletes();
    $table->timestamps();
});
```

```php
Schema::create('customers', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('email')->unique();
    $table->string('phone')->nullable();
    $table->string('address');
    $table->date('birthdate');
    $table->timestamps();
});
```

```php
Schema::create('categories', function (Blueprint $table) {
    $table->id();
    $table->string('name')->unique();
    $table->text('description')->nullable();
    $table->boolean('active')->default(true);
    $table->softDeletes();
    $table->timestamps();
});
```

```php
Schema::create('orders', function (Blueprint $table) {
    $table->id();
    $table->foreignId('customer_id')->constrained('customers')->onDelete('cascade');
    $table->foreignId('product_id')->constrained('products')->onDelete('cascade');
    $table->integer('quantity')->default(1);
    $table->decimal('total_price', 10, 2);
    $table->dateTime('order_date');
    $table->string('status')->default('pending');
    $table->text('notes')->nullable();
    $table->softDeletes();
    $table->timestamps();
});
}
```

Aquí como vemos creamos tablas de id, string, decimal, date, etc para rellenar los datos

3)El código que usaron para hacer las relaciones.

```php
$table->foreignId('customer_id')->constrained('customers')->onDelete('cascade');
$table->foreignId('product_id')->constrained('products')->onDelete('cascade');
```

Estas dos líneas están creando relaciones foráneas (foreign keys) en una tabla y aquí va cada parte explicada:

`$table->foreignId('customer_id')`

- Crea una columna llamada `customer_id` de tipo **BIGINT sin signo**.
- Es la convención que Laravel usa para claves foráneas (`nombre_tabla_singular_id`).

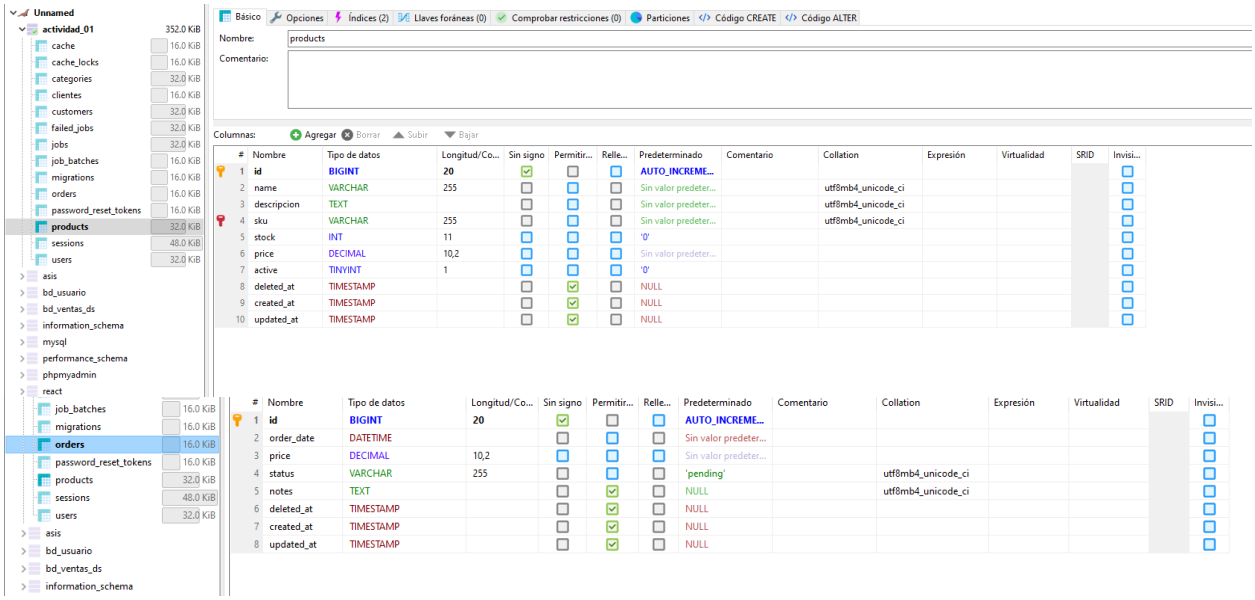
`$table->constrained('customers')`

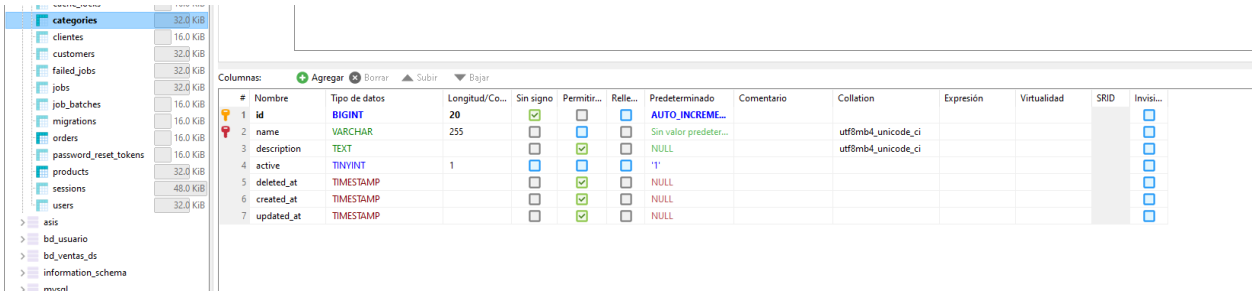
- Le dice a Laravel que `customer_id` es una **clave foránea** que hace referencia a la columna `id` de la tabla **`customers`**.
- Laravel por defecto asume que se relaciona con el campo `id` si no se especifica lo contrario.

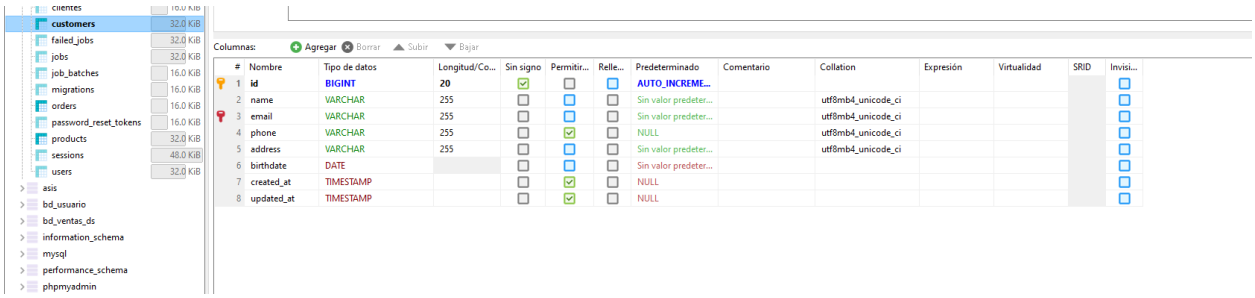
$table `->onDelete('cascade')`

- Significa que **si se elimina un cliente**, **todos los registros relacionados** en esta tabla (por ejemplo, sus pedidos) **también se eliminarán automáticamente**.
- Esta es una regla de integridad referencial definida en la base de datos.

4)Captura y/o evidencia de que existen las tablas y sus respectivos campos en HEIDISQL.







Aquí como vemos al momento de migrate a los tablas creadas se agregan al HEIDISQL y vemos todos agregado en el visual

5)Detallar el procedimiento mencionado líneas arriba de cada tabla.

$table->string(name)

- Crea una columna de tipo **cadena de texto corta** (hasta 255 caracteres).
- Guarda el nombre del producto

$table->text('descripcion')

- Guarda la **descripción del producto**, sin límite fijo de caracteres.

$table->string('sku')->unique();

- Crea una cadena llamada `sku` (por ejemplo, código de inventario).
- el unique para que no se repita una tabla

$table->integer('stock')->default(0);

- Crea un stock y el default deja el predeterminado n 0 para que se pueda modificar

$table->decimal (“Price”, 10.2);

- Hasta **10 dígitos en total**,
- **2 después del punto decimal**.

$table->boolean('active')->default(false);

- Crea una columna **booleana** (true/false).
- Indica si el producto está **activo o no**.
- Por defecto está en **false (inactivo)**.

$table->softDeletes();

- Agrega la columna `deleted_at` para el **borrado lógico**.
- Elimina "virtualmente" sin quitar el registro de la base de datos.