**UNIVERSIDAD DE EL SALVADOR**

**FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE**

**DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**

****

**PRÁCTICA N° 3**

**ASIGNATURA:**

**Técnicas De Programación En Internet**

**DOCENTE:**

**Ing. Xenia Ivette Peñate Godoy**

**INTEGRANTES:**

**Calderón López, Fátima del Rosario    Carnet: CL16006**

**Ponce Quijada, Jessica Lissette          Carnet: PQ16001**

**PRÁCTICA 3**

Creación de una aplicación haciendo uso de un framework

En la actualidad, se cuenta con múltiples herramientas que facilitan el desarrollo web, podemos disponer de gran cantidad de librerías, así como de plugins enmarcados dentro de lo que se denominan Frameworks.

Podemos definir un Framework como un conjunto de módulos que permiten, o tienen por objetivo, el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas para que nosotros las usemos directamente.

Por tal razón, un Framework se encarga de facilitar el desarrollo, estandarizar las páginas y son de mucha utilidad para el desarrollo de aplicaciones web más escalables y sencillas de mantener, lo que nos permite también ahorrar tiempo.

Para realizar esta práctica hemos utilizado las siguientes herramientas:

**Framework Django:**

Django es un framework web de código abierto, escrito en Python, que se ajusta en gran medida al conocido patrón de diseño MVC (Modelo Vista Controlador) el cual permite crear aplicaciones web extensibles, escalables y fáciles de mantener.

**Entorno de desarrollo PyCharm:**

Es un [entorno de desarrollo integrado](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) utilizado en [programación de computadoras](https://en.wikipedia.org/wiki/Computer_programming) , específicamente para el lenguaje [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) . Está desarrollado por la empresa [checa](https://en.wikipedia.org/wiki/Czech_Republic)[JetBrains](https://en.wikipedia.org/wiki/JetBrains). Proporciona análisis de código, un depurador gráfico, un probador de unidades integrado, integración con [sistemas de control de versiones](https://en.wikipedia.org/wiki/Revision_control) y admite el desarrollo web con [Django](https://en.wikipedia.org/wiki/Django_(web_framework)).

**Lenguaje de programación Python:**

Python es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [interpretado](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)) cuya filosofía hace hincapié en la legibilidad de su [código](https://es.wikipedia.org/wiki/Codigo_fuente). Se trata de un lenguaje de programación [multiparadigma](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n#Paradigma_de_programaci%C3%B3n), ya que soporta parcialmente la [orientación a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), programación imperativa y, en menor medida, [programación funcional](https://es.wikipedia.org/wiki/Programaci%C3%B3n_funcional). Es un [lenguaje interpretado](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_interpretado), [dinámico](https://es.wikipedia.org/wiki/Tipado_din%C3%A1mico) y [multiplataforma](https://es.wikipedia.org/wiki/Multiplataforma).

**PROCEDIMIENTO**

Primero debemos instalar Django, para ello tenemos que tener instalado previamente Python 3 en nuestra máquina, para Instalar Django utilizamos el comando:

pip install Django==3.2.3

para ver la que lo tenemos instalado ejecutamos el comando:

py -m django --version

creamos el proyecto

Para ello tenemos que ubicarnos en el directorio donde tenemos nuestros proyectos con django y ejecutamos el comando

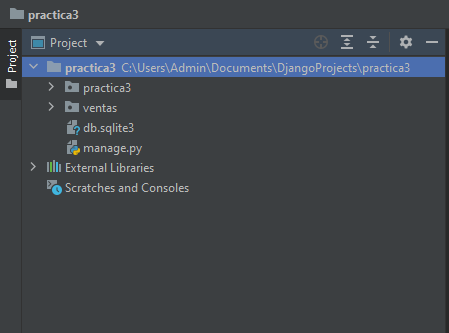
py –m django startproyect practica3

Podemos abrir el proyecto en el IDE en nuestro caso usamos Pycharm.

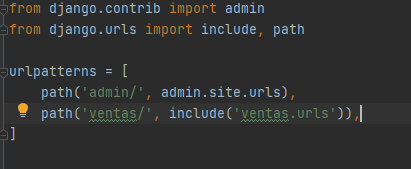
En pycharm abrimos una terminal y ejecutamos el comando:

python manage.py startapp ventas

Esto nos crea una aplicación dentro de nuestro proyecto principal, cada aplicación que usted escribe en Django consiste en un paquete de Python que sigue una determinada convención. Django tiene una utilidad que genera automáticamente la estructura básica de directorios de una aplicación, por lo que usted puede centrarse en la escritura de código en lugar de crear directorios.

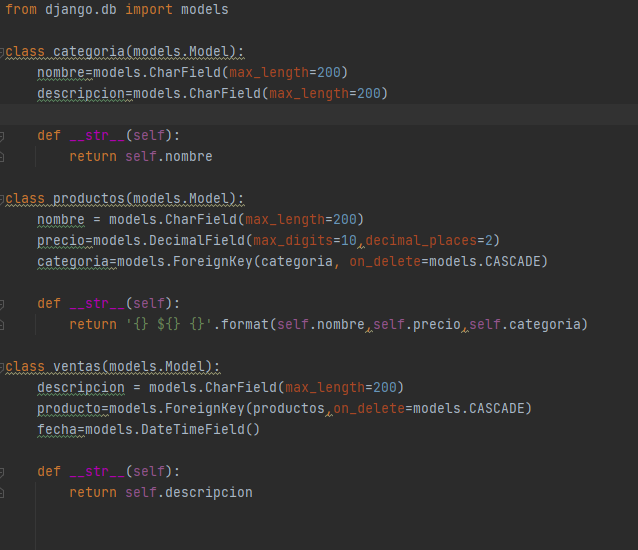


Agregamos nuestra aplicación al archivo practica3/urls.py



Vamos a empezar trabajando en el modelo, abrimos el archivo ventas\models.py

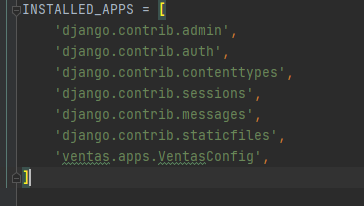
Y lo editamos como sigue:



Ahora que tenemos el modelo realizamos las migraciones con el comando:

python manage.py migrate

Para incluir la aplicación en nuestro proyecto necesitamos agregar una referencia a su clase de configuración en la configuración INSTALLED\_APPS. La clase VentasConfig está en el archivo ventas/apps.py por lo que su ruta punteada es 'ventas.apps.VentasConfig'. Edite el archivo *practica3/settings.py* y agregue la ruta punteada a la configuración INSTALLED\_APPS. Se verá así:



Ahora hay que hacer las migraciones con el comando:

python manage.py makemigrations ventas

Al ejecutar makemigrations, usted le indica a Django que ha realizado algunos cambios a sus modelos (en este caso, ha realizado cambios nuevos) y que le gustaría que los guarde como una *migración*.

Ahora creamos un superusuario para el panel de administración con el comando:

python manage.py createsuperuser

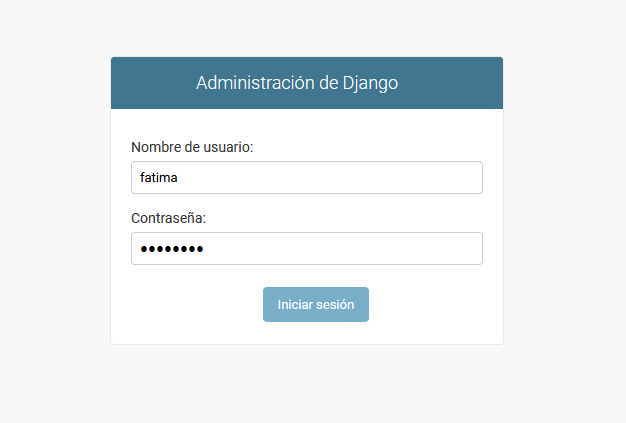
nos pide que ingresemos un nombre, un email, y que creemos una contraseña.

Iniciamos el servidor con el comando:

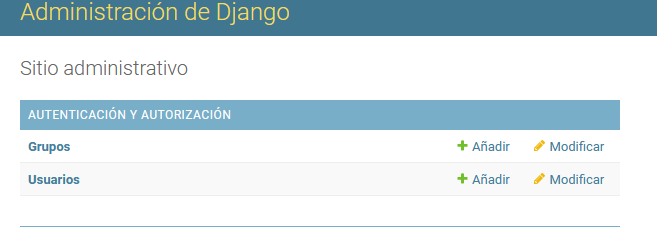
python manage.py runserver

Y nos dirigimos a la ruta <http://127.0.0.1:8000/admin/>

Tendremos que ingresar el usuario y contraseña en este caso el usuario es “fatima” y la contraseña es “fatima97”

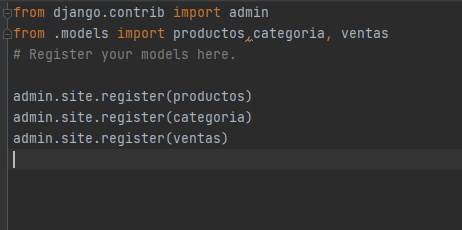


Y ya estamos dentro del panel de administración de la aplicación



Podemos crear grupos y usuarios

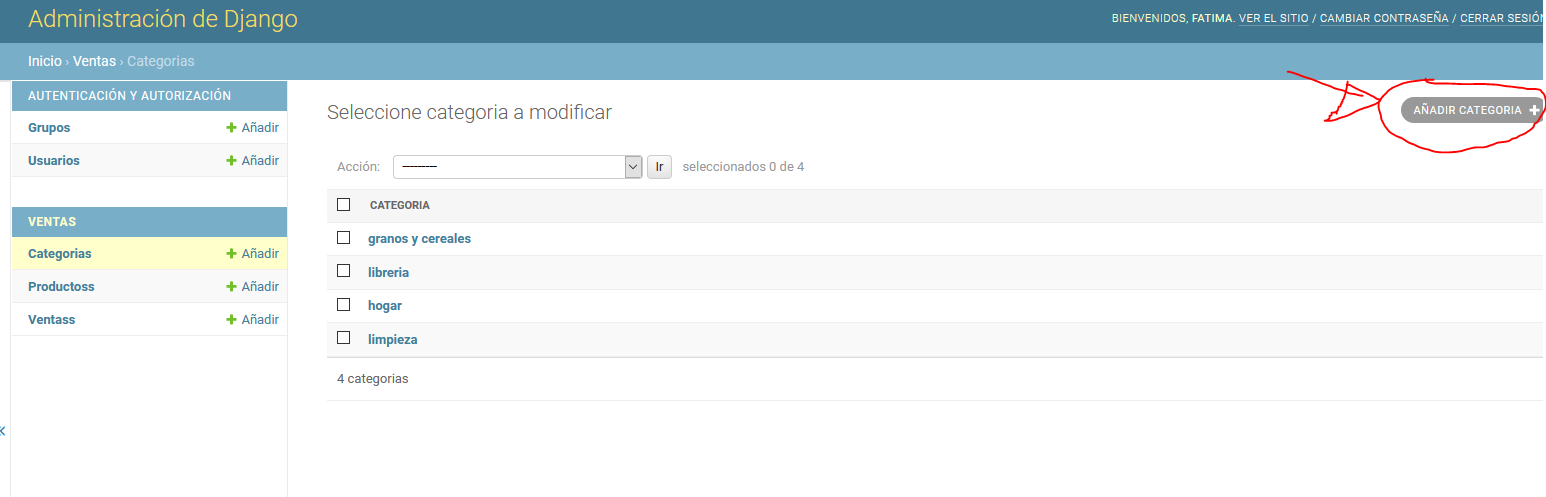
Vamos a editar el archivo ventas/admin.py



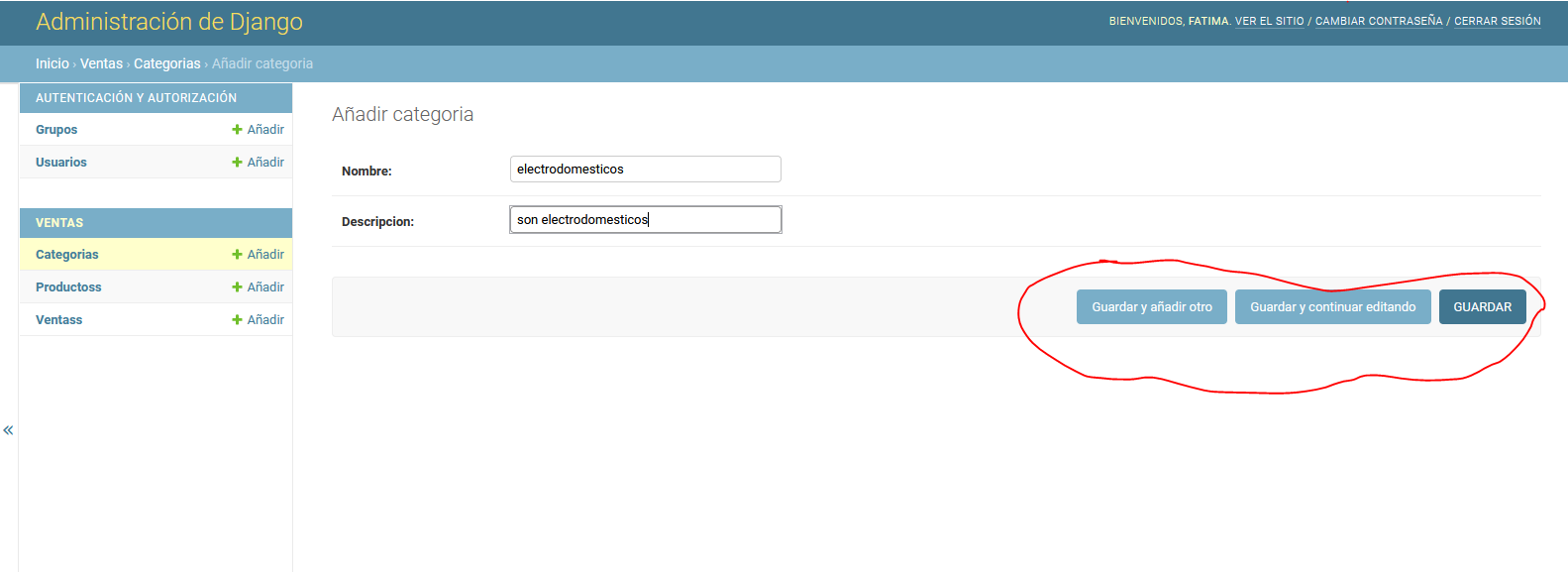
Recargamos el panel de administración y ahora ya podemos hacer las operaciones crud a nuestros modelos:



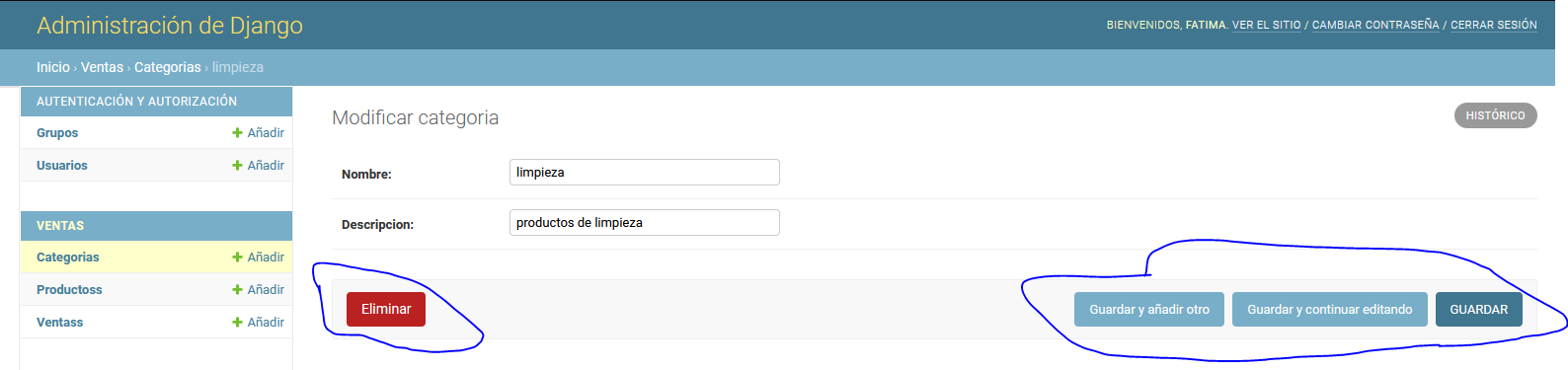
Si damos click en categorías podemos agregar categorías:



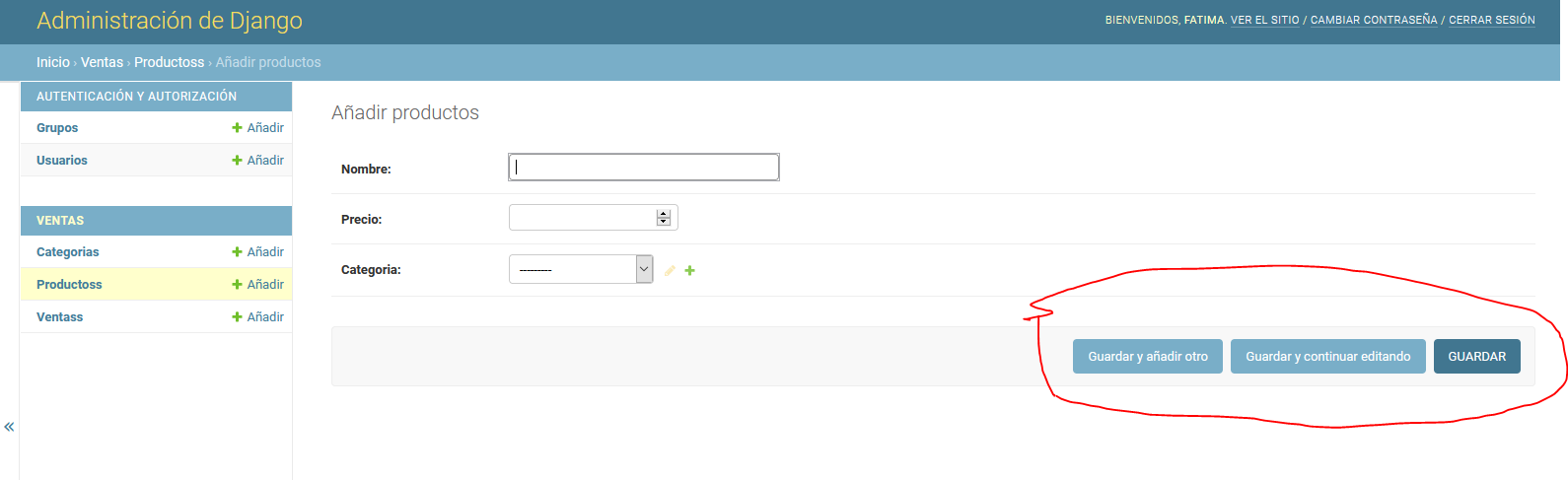
Podemos crear una nueva categoría



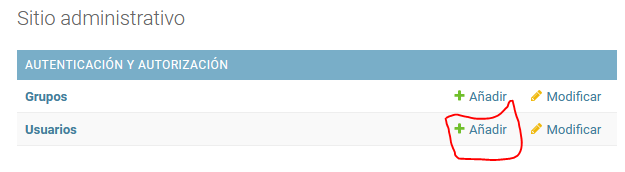
Podemos modificar o eliminar una categoría

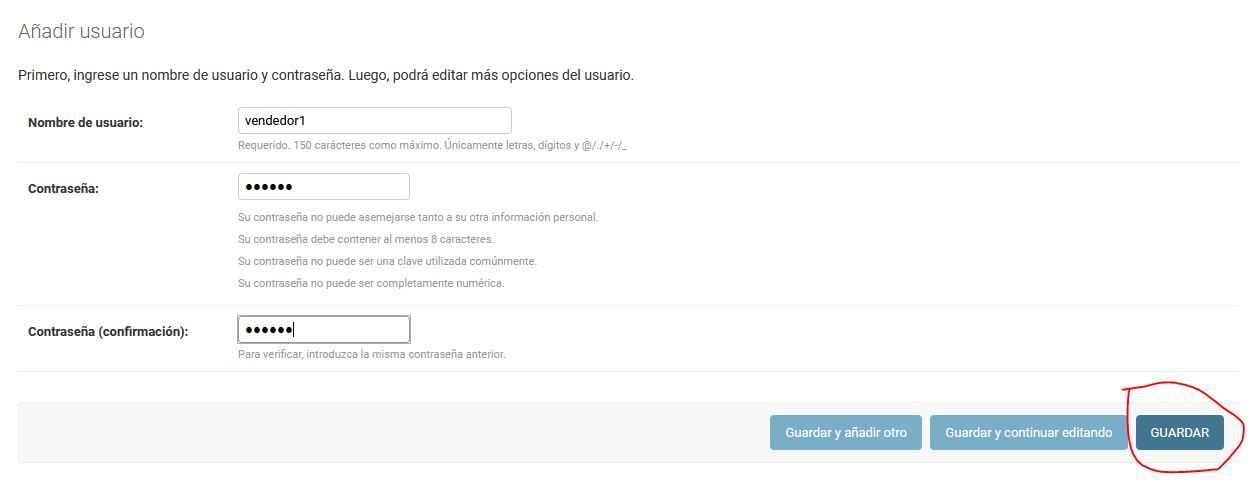


Y así podemos también gestionar productos y ventas:

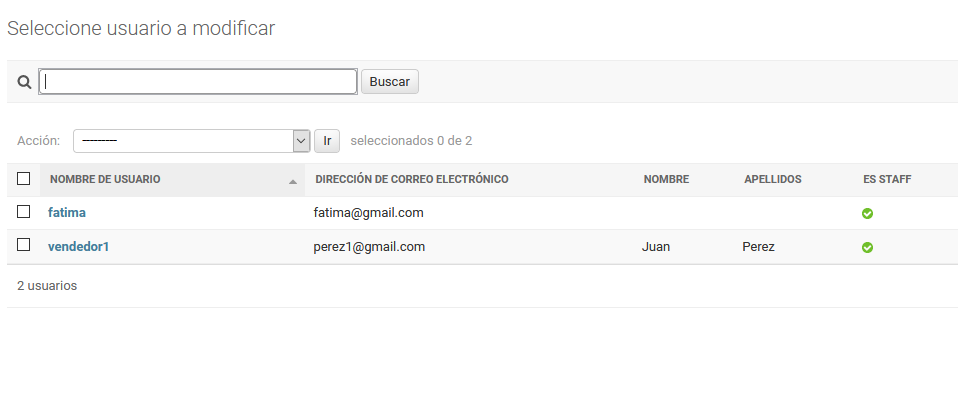


Ahora vamos a crear un usuario para que pueda solo hacer ventas

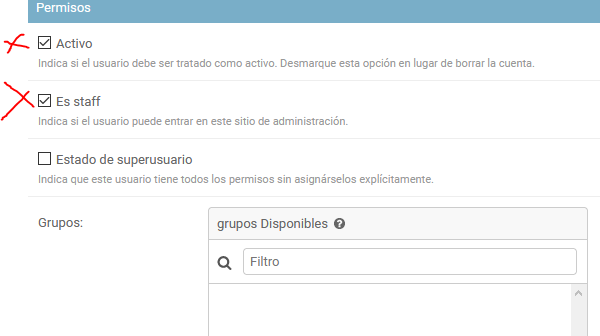




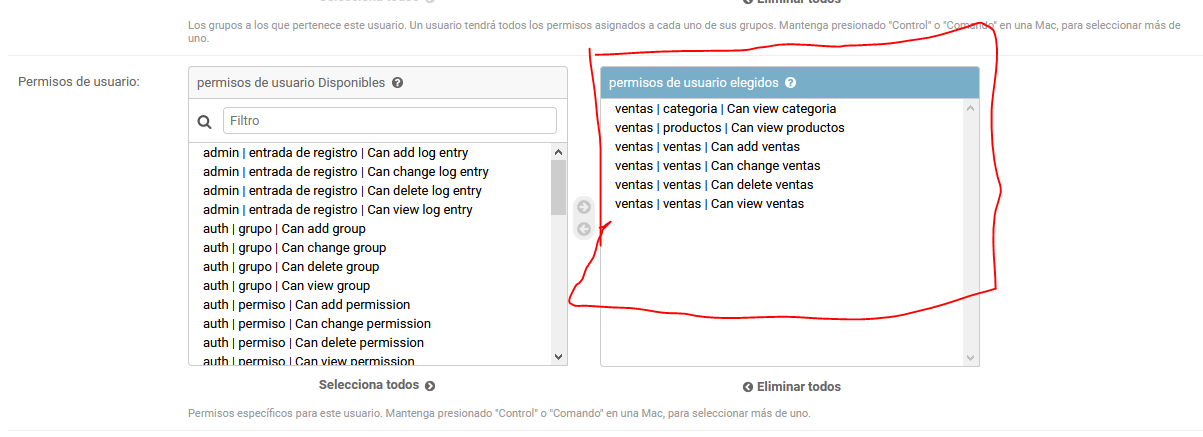
Vamos a darle permisos a nuestro usuario



Debemos marcar las siguientes casillas



Y dar los siguientes permisos



Guardamos todos los cambios.

Ahora cerramos sesión y vamos a iniciar sesión con el usuario “vendedor1” y con la contraseña “13052021v1”.



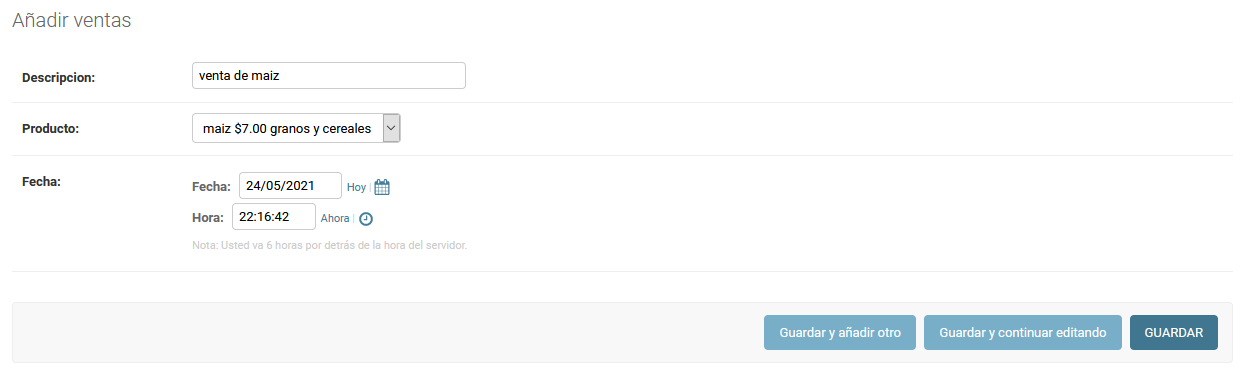
Ahora nuestro usuario “vendedor1” puede hacer ventas, pero solo podrá ver las categorías y los productos.



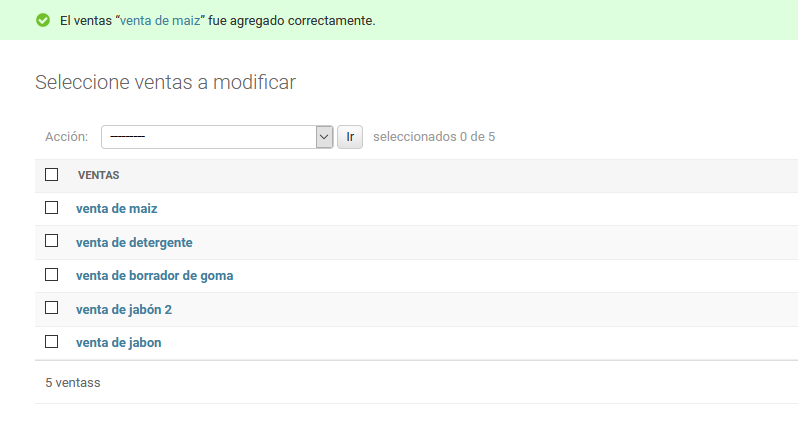




Hacemos una venta



Al darle en guardar nos aparecerá un mensaje que la venta fue hecha y aparecerá en la lista:



Enlace del repositorio de GitHub:

<https://github.com/CL16006/practica3-unidad3-tpi>