

Universidad Salesiana de Bolivia

Ingeniería de Sistemas



Reingenieria de Software

Estudiante: Ruben Mamani Tito

Docente: ing. Freddy Felix Gutierrez Chavez

Materia: Ingeniería de Software II

GESTIÓN – 2022

La paz-Bolivia

Reingenieria de Software

El sistema trata de una tienda de venta de productos online el cual tiene los siguientes módulos:

Servicios. - Modulo donde se encuentra los servicios de la empresa, como ser la atención al cliente, servicio de entrega del producto.

Pedidos. - Enlista la información de todos los productos a disposición para la venta.

Contacto. – Información de cómo se puede contactar el cliente con la empresa ya sea directamente o mediante las redes sociales.

Blog. – Modulo donde los clientes pueden opinar acerca los productos o la empresa.

El proceso el cual se realizará la reingeniería es el módulo de PEDIDOS.

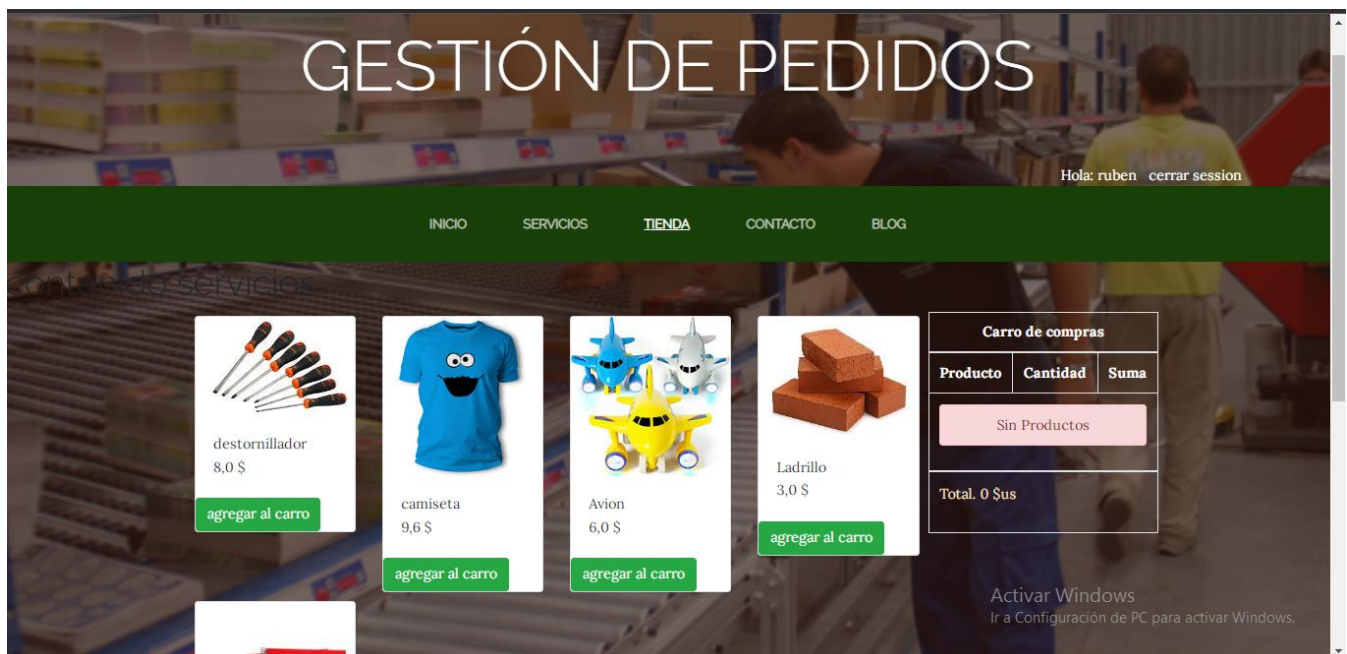
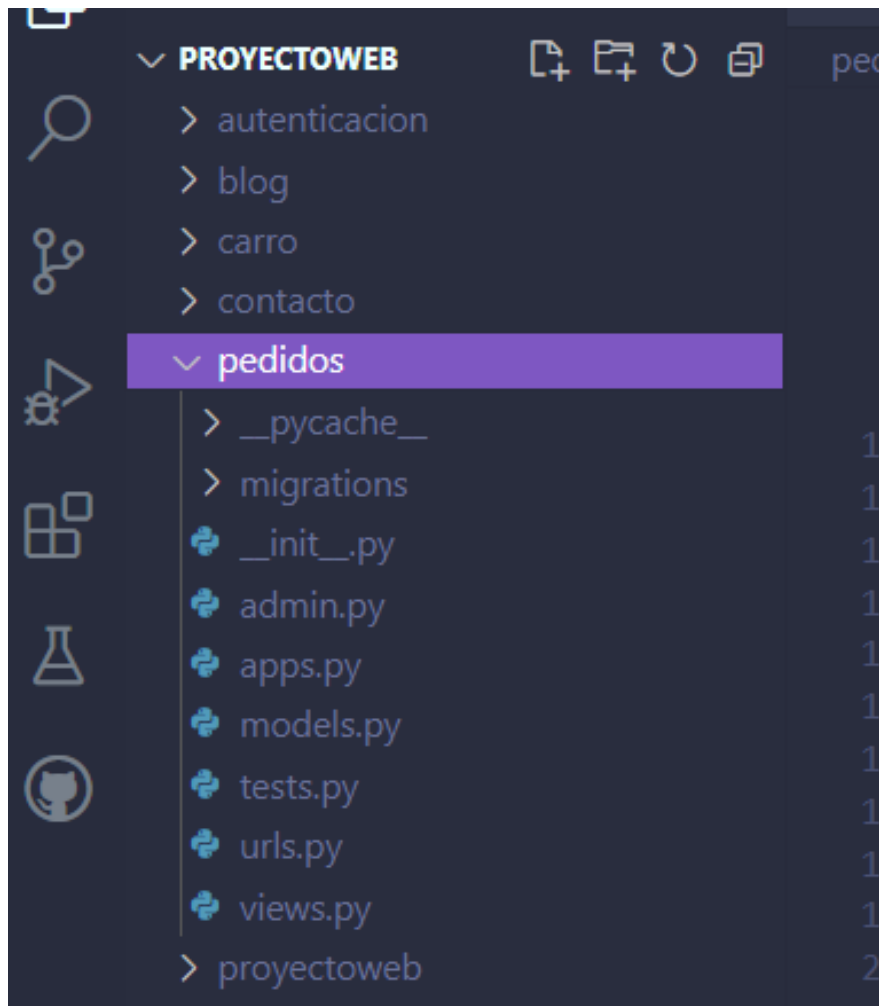


Foto del sistema

El sistema está desarrollado en el framework Django que se basa en el lenguaje de programación Python y usa el patrón MTV (Modelo, vista, plantilla) y tiene la siguiente estructura:



Para esta aplicación de software se realizará una reestructuración de código en el módulo de PEDIDOS, específicamente en el archivo de models.py.

El archivo models.py presenta un problema de código fuente, el cual se considera que existen líneas de código y funciones que no son utilizadas para el proceso de pedidos de productos.

Código fuente original

Las funciones que no son utilizadas son: class meta, el cual esa función ya es caduco para el sistema.

También se pudo observar que no se hace uso de ciertas importaciones

```
from django.db import models
from django.contrib.auth import get_user_model
from tienda.models import Producto
from django.db.models import F, Sum, FloatField
# Create your models here.
User=get_user_model()

class Pedido(models.Model):
    user=models.ForeignKey(User, on_delete=models.CASCADE)
    created_at=models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return self.id
    @property
    def total(self):
        return self.lineapedido_set.aggregate(
            total=Sum(F("precio")*F("cantidad"), output_field=FloatField())
        )["total"]
    class Meta:
        db_table='pedidos'
        verbose_name='pedido'
        verbose_name_plural='pedidos'
        ordering=['id']
class LineaPedido(models.Model):
    user=models.ForeignKey(User, on_delete=models.CASCADE)
    producto=models.ForeignKey(Producto, on_delete=models.CASCADE)
    pedido=models.ForeignKey(Pedido, on_delete=models.CASCADE)
    cantidad=models.IntegerField(default=1)
    def __str__(self):
        return f'{self.cantidad} unidades de {self.producto.nombre}'
    class Meta:
        db_table='lineapedidos'
        verbose_name='Linea pedido'
        verbose_name_plural='Lineas Pedidos'
        ordering=['id']
```

Código fuente reestructurada

Para reestructurar el código fuente, se tuvo que eliminar y/o modificar las funciones y código que no se usaban.

```
from django.db import models
from django.contrib.auth import get_user_model
# Create your models here.
User=get_user_model()

class Pedido(models.Model):
    user=models.ForeignKey(User, on_delete=models.CASCADE)
    created_at=models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return self.id
    @property
    def total(self):
        return self.lineapedido_set.aggregate(
            total=Sum(F("precio")*F("cantidad"), output_field=FloatField())
        )["total"]

class LineaPedido(models.Model):
    user=models.ForeignKey(User, on_delete=models.CASCADE)
    producto=models.ForeignKey(Producto, on_delete=models.CASCADE)
    pedido=models.ForeignKey(Pedido, on_delete=models.CASCADE)
    cantidad=models.IntegerField(default=1)
    def __str__(self):
        return f'{self.cantidad} unidades de {self.producto.nombre}'
```