UNIVERSIDAD CONTINENTAL

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

PROYECTO

“Implementación de una Plataforma Web Ecommerce para la empresa indiePrints”

PRESENTADO POR:

APELLIDOS Y NOMBRES	CÓDIGO
Hilario Caso Hieyson Steven	77486405
Angela Paola Torres Torres	75026225
Güere Casas Jose Manuel	72047656

ASESOR:

JORGE ALFREDO GUEVARA JIMENEZ

HUANCAYO – PERÚ

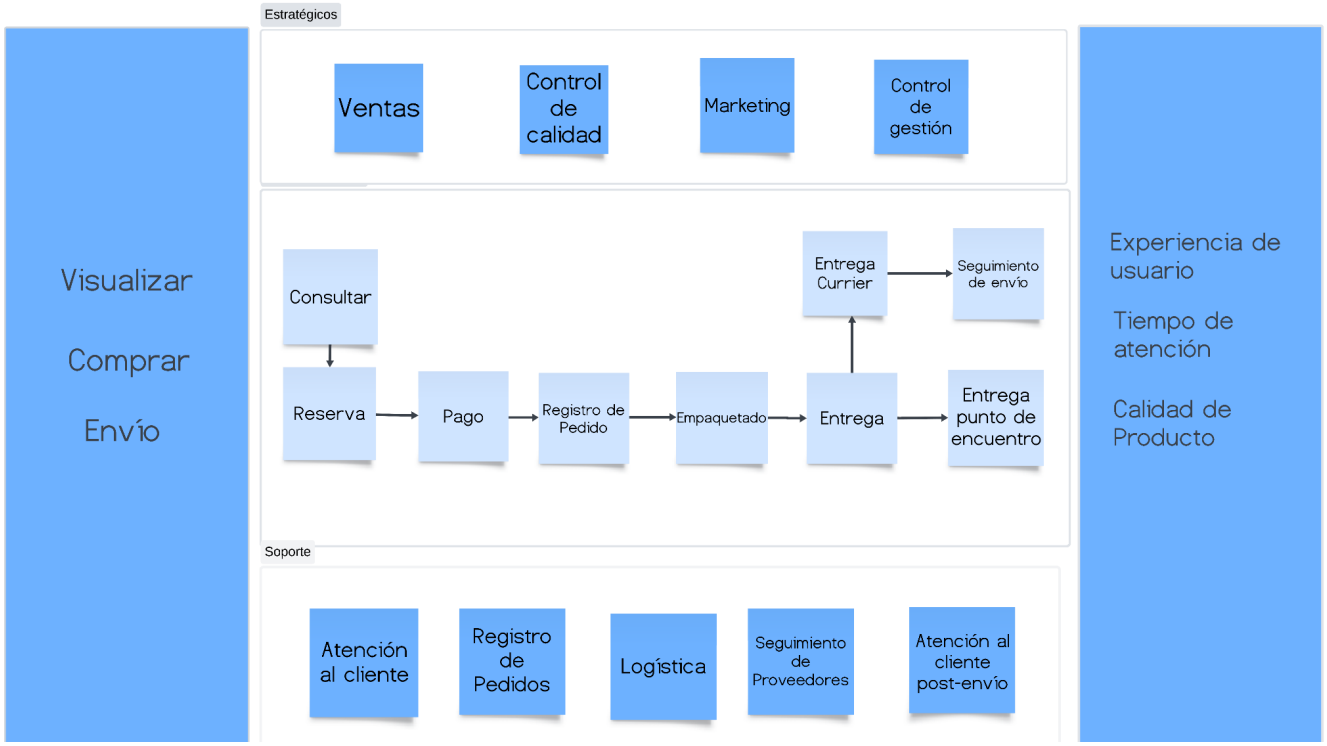
2023

ÍNDICE

PORTADA **1**
1. Mapa de procesos **3**
2. Reseña de la Organización **3**
3. Flujograma de la cadena de Valor **4**
4. Modelo incremental **4**
4.1 Requerimientos Incremento 1: Visualizar Catálogo 4
5. Plan de unidades de programación **5**
6. Modelo Entidad -Relación **6**
8. Arquitectura de TI de Hardware y software **8**
9. Planificación de Pruebas **9**
10. Ejecución de plan de pruebas :Incremento 1 **10**
Caso de uso : Registrar Producto 10
Caso de uso : Registrar Talla 11
Caso de uso : Material 12
Caso de uso : Color 13
Caso de uso : Categoría 14

1. Mapa de procesos

Misión	Visión
Nuestra misión es proporcionar a nuestros clientes una experiencia de compra excepcional al ofrecer una amplia gama de polos de bandas indie .	Nuestra visión es ser la tienda virtual de merchandising más grande de Huancayo, ofreciendo una amplia variedad de productos.

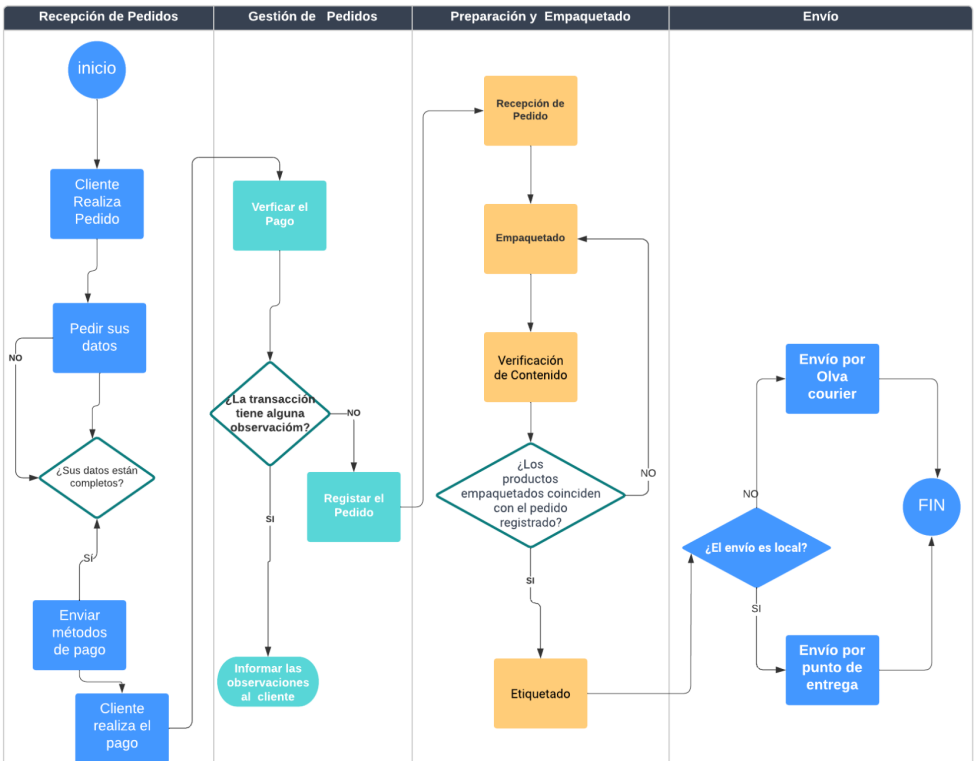


2. Reseña de la Organización

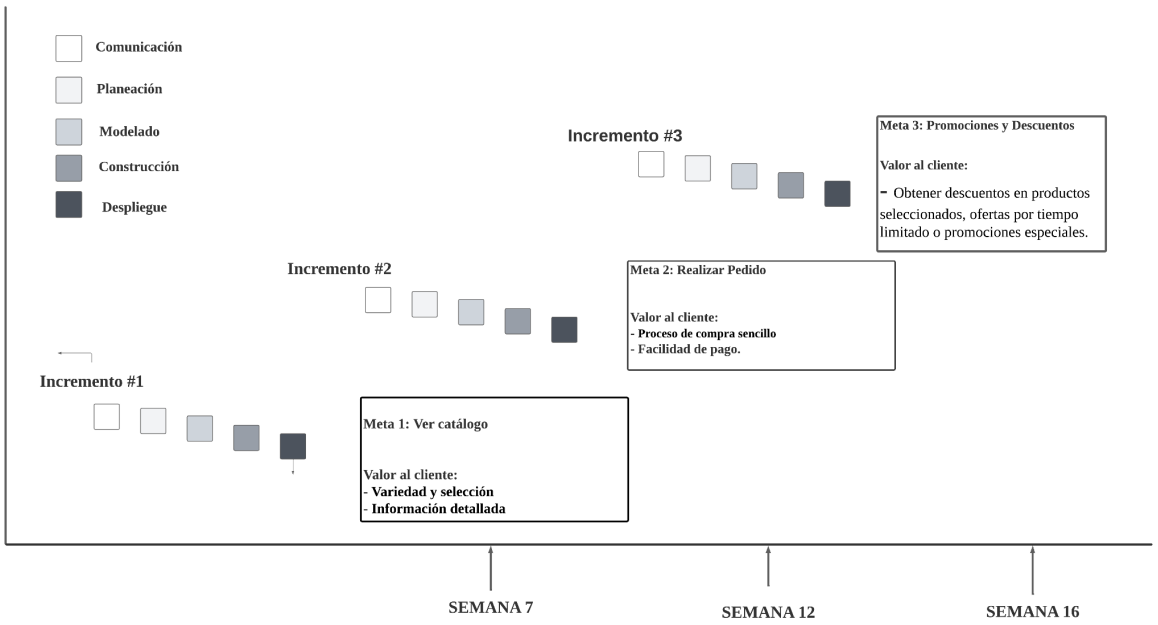
2.1 Descripción

Nuestra empresa de e-commerce se especializa en la venta de polos de bandas indie y polos personalizados.
Nos enorgullece ofrecer una amplia selección de prendas de alta calidad que capturan el espíritu creativo y único de las bandas indie, así como la posibilidad de personalizar polos según las preferencias individuales de nuestros clientes.

3. Flujograma de la cadena de Valor



4. Modelo incremental



4.1 Requerimientos Incremento 1: Visualizar Catálogo

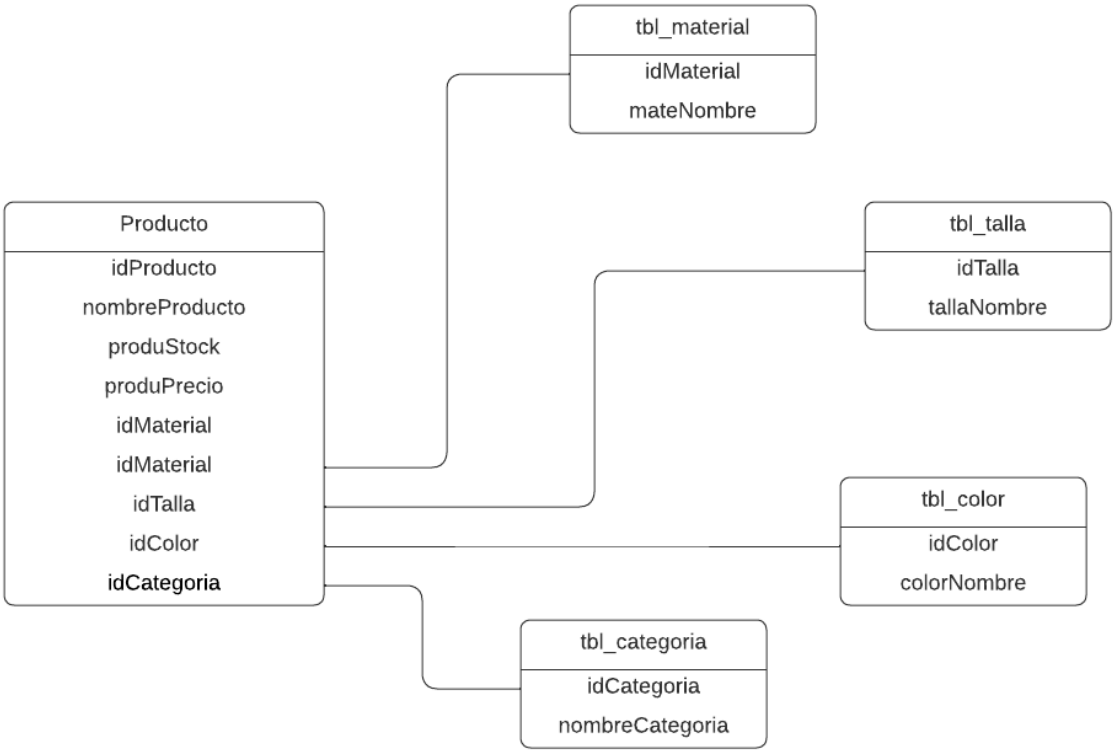
#Incremento	R. Funcionales	R. No Funcionales
1 Catálogo	Mostrar Productos Realizar Búsquedas Filtrar Búsquedas por categoría	Compatibilidad multiplataforma Usabilidad y experiencia de usuario Registro de errores y seguimiento Django , Pycharm Mongodb y SQLAlchemy

5. Plan de unidades de programación

Requerimiento	Incremento	tipo clase	Nombre de clase	Responsable	Fecha de inicio	Fecha Fin
Mostrar Producto	1	Modelo	M_mostrar_producto	Hilario Caso	14/9/2023	17/9/2023
		Vista	V_mostrar_producto	Güere Casas	18/9/2023	21/9/2023
		Controlador	C_mostrar_producto	Torres Torres	22/9/2023	25/9/2023
Realizar Busqueda	1	Modelo	M_Busqueda_producto	Hilario Caso	26/9/2023	29/9/2023
		Vista	V_Busqueda_producto	Güere Casas	30/9/2023	3/10/2023
		Controlador	C_Busqueda_producto	Hilario Caso	4/10/2023	7/10/2023

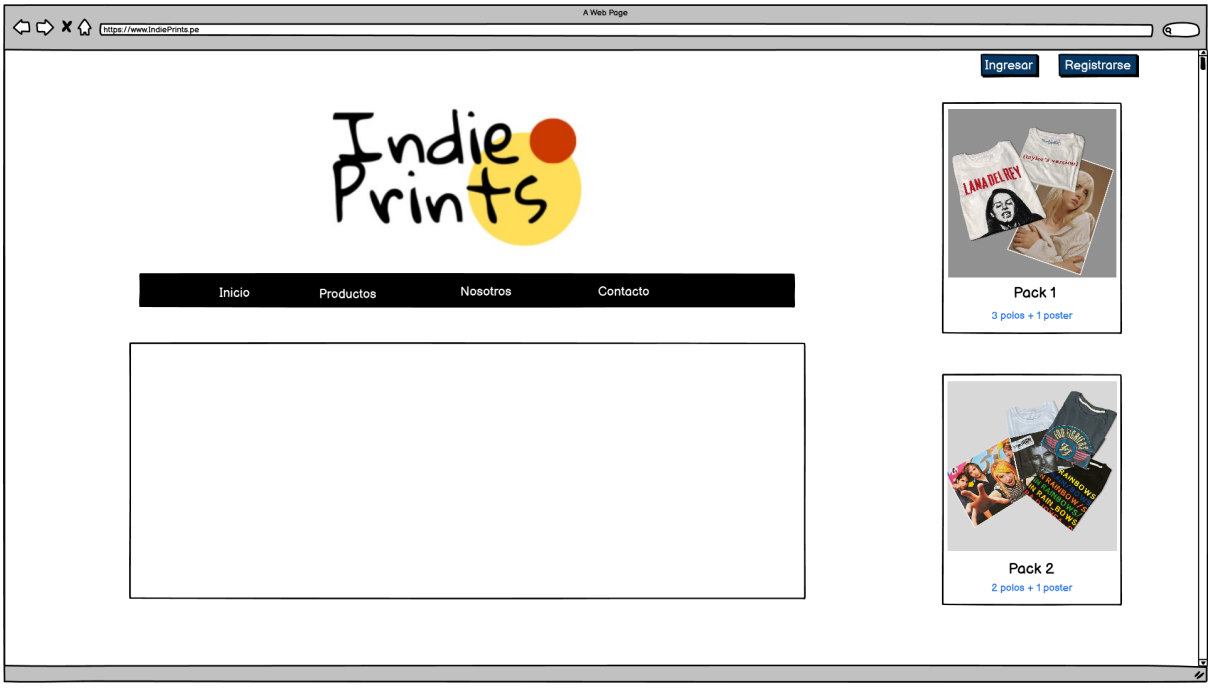
Filtros Categoría	1	Modelo	M_Filtro_producto	Hilario Caso	8/10/2023	11/10/2023
		Vista	V_Filtro_producto	Torres Torres	12/10/2023	15/10/2023
		Controlador	C_Filtro_producto	Güere Casas	16/10/2023	19/10/2023

6. Modelo Entidad -Relación



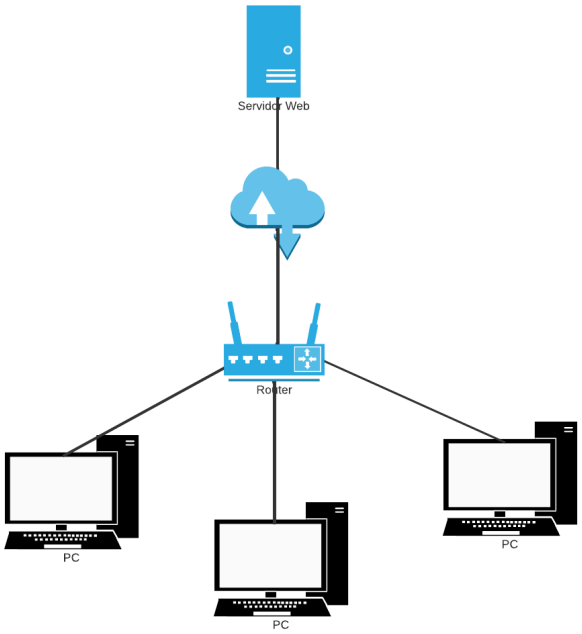
7.Mockups

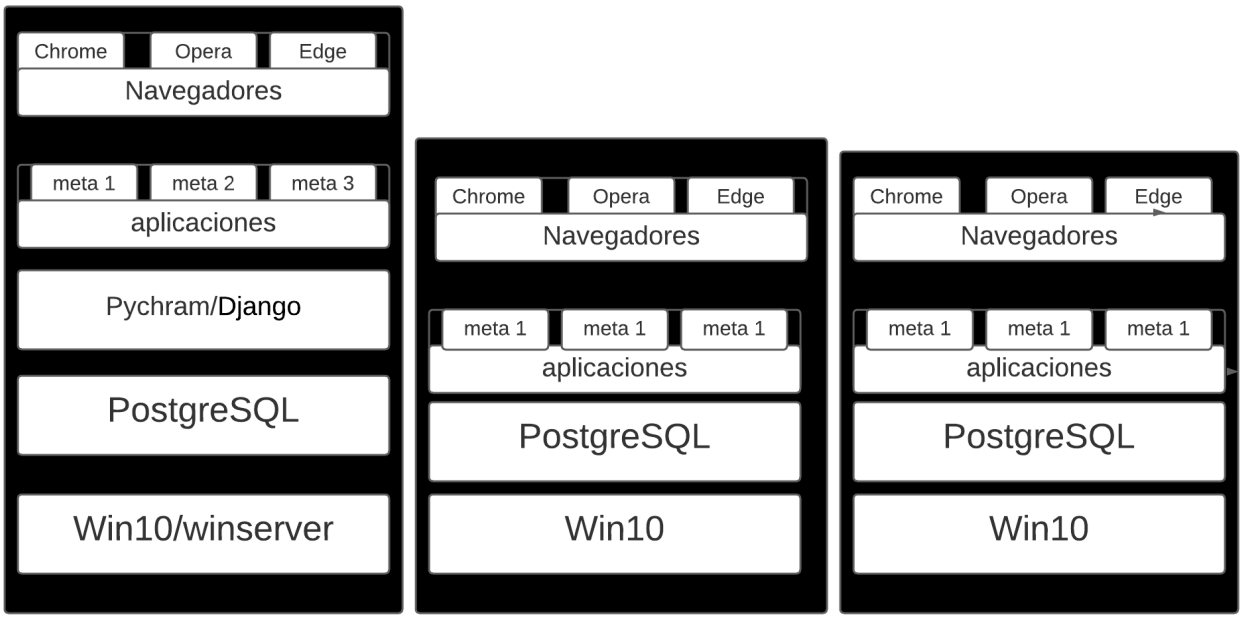
Pantalla de Inicio



Pantalla que muestra los productos , además se puede buscar por el nombre o filtrar las búsquedas por categoría.

8. Arquitectura de TI de Hardware y software





Meta1: Catálogo
Meta2:Realizar Pedido
Meta3: Descuentos y Promociones

9. Planificación de Pruebas

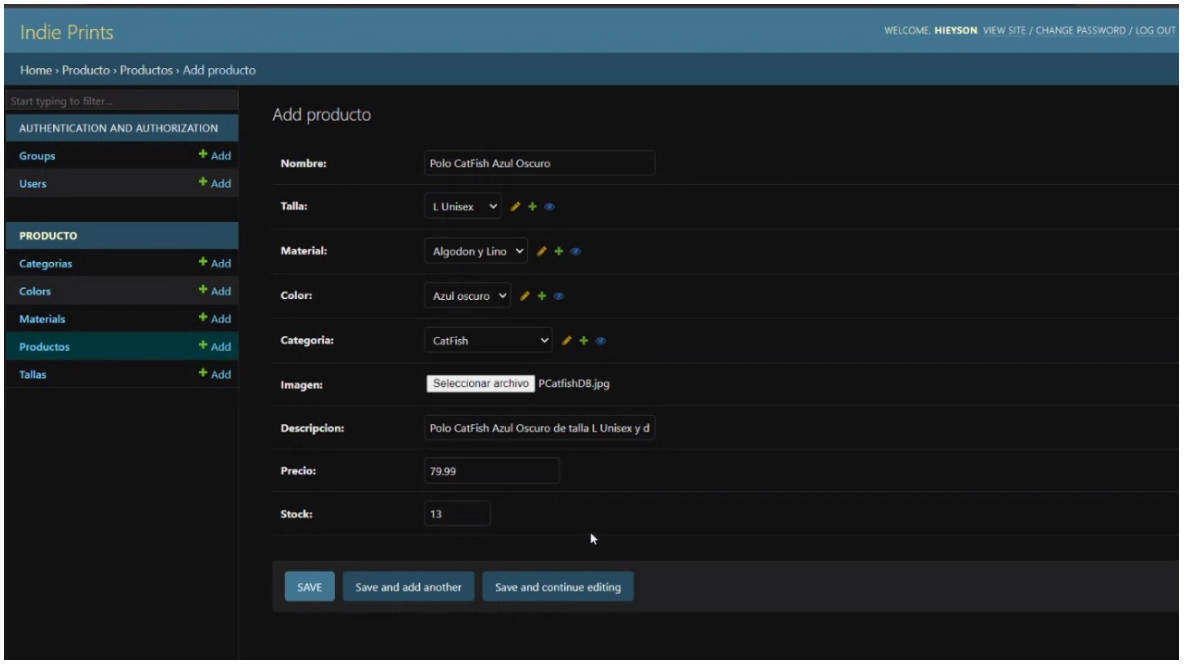
Casos de Uso	Datos de entrada Válidos	Resultado Esperado normal	Datos de entrada no válidos	Resultado Esperado de datos anómalos	Resultado final 1	2	3
Registrar Producto	ID	Producto registrado correctamente	ID duplicado Datos en blanco	Error en registrar producto	✓	✓	✓
	nombreProducto		Datos en blanco				
	Talla		Datos en blanco				
	Material		Datos en blanco				
	Color		Datos en blanco				
	Categoría		Datos en blanco				
	Stock		Datos en blanco				
Registrar Talla	ID	Producto registrado correctamente	ID duplicado	Error en registrar producto	✓	✓	✓
	nombreTalla		Datos en blanco				
Registrar Material	ID	Producto registrado correctamente	ID duplicado	Error en registrar producto	✓	✓	✓
	nombreMaterial		Datos en blanco				
Registrar Color	ID	Producto registrado correctamente	ID duplicado	Error en registrar producto	✓	✓	✓
	nombreColor		Datos en blanco				
Registrar Categoría	ID	Producto registrado correctamente	ID duplicado	Error en registrar producto	✓	✓	✓
	nombreCategoria		Datos en blanco				

Mostrar Producto		Visualizar el producto			✓	✓	✓
Búsqueda por nombre	nombreProducto		Datos en blanco	El producto no existe			
Filtro por categoría	Categoría	Filtrar	Datos en blanco		✓	✓	✓

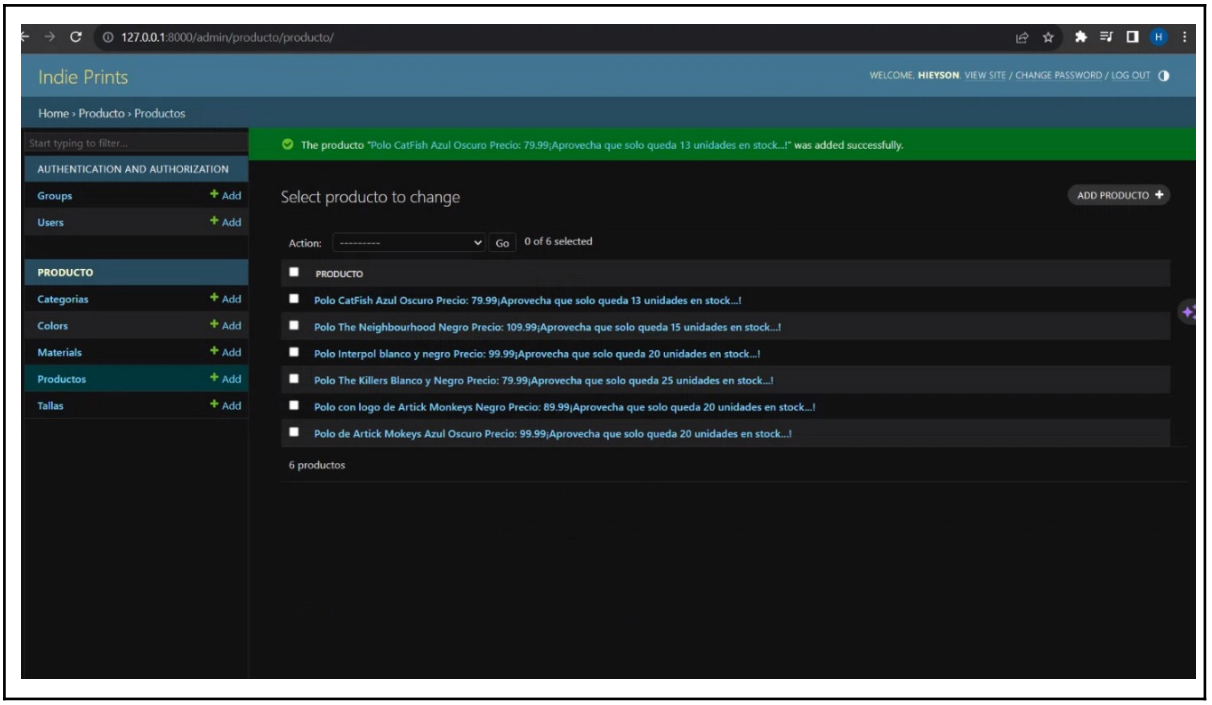
10. Ejecución de plan de pruebas :Incremento 1

Caso de uso : Registrar Producto

Datos de entrada Válidos

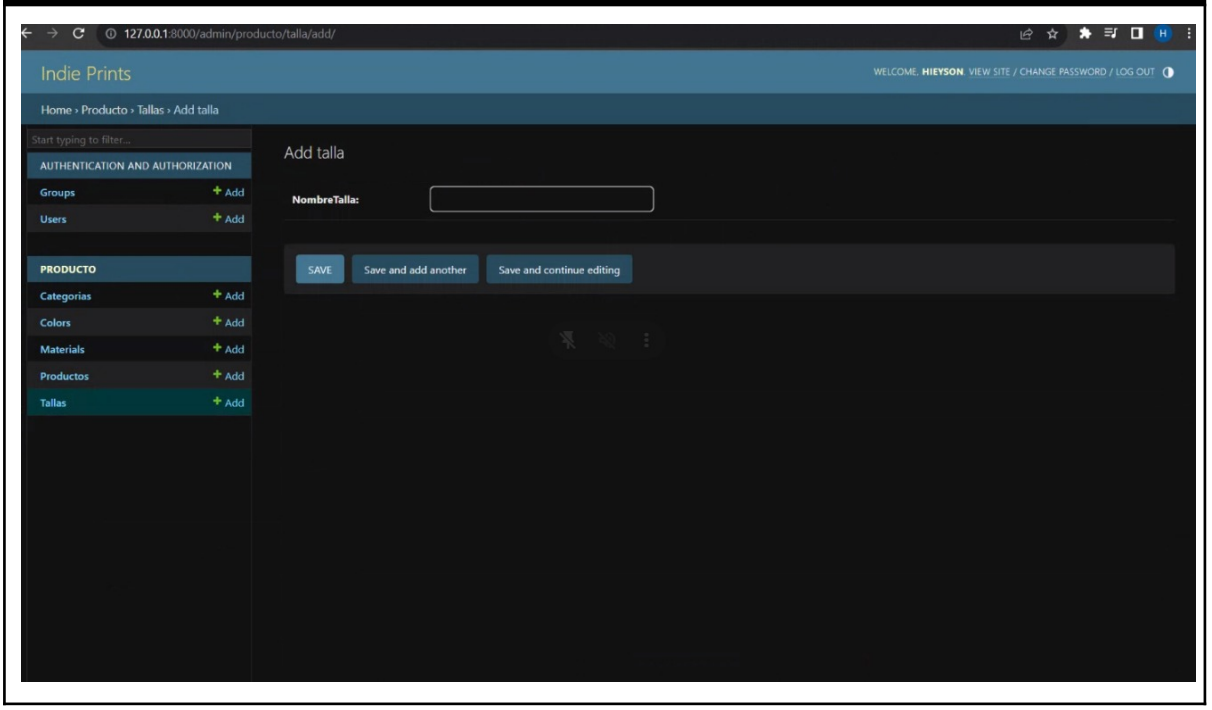


Resultado Esperado normal

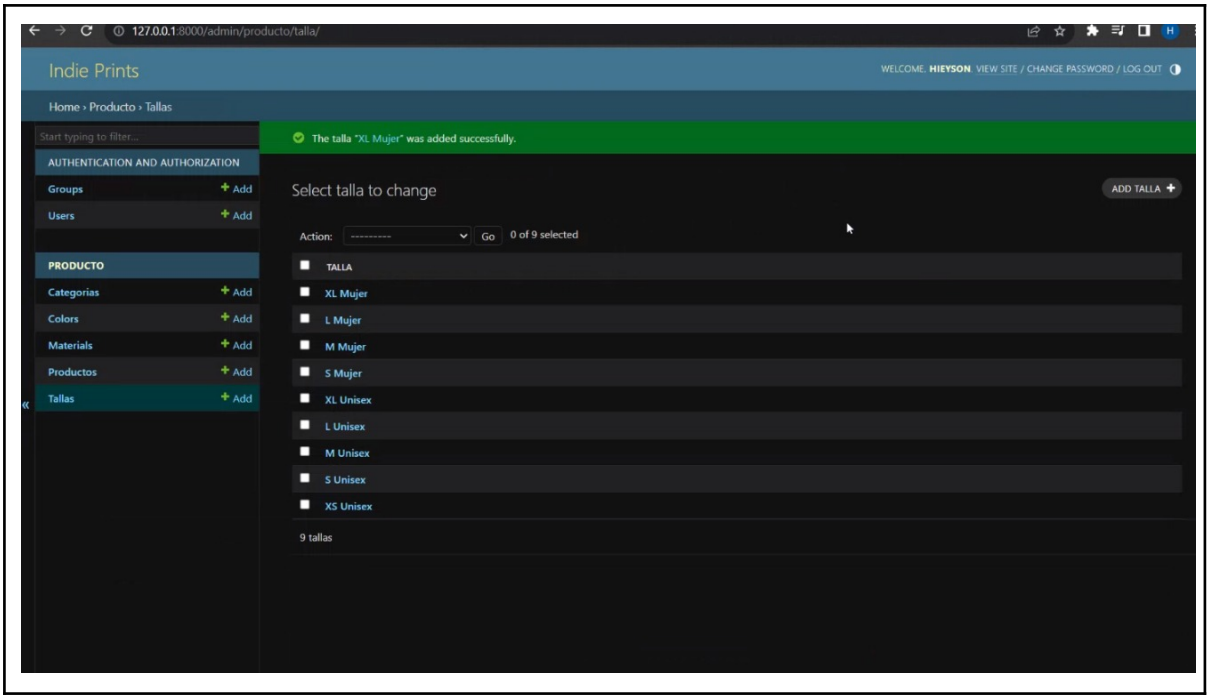


Caso de uso : Registrar Talla

Datos de entrada Válidos

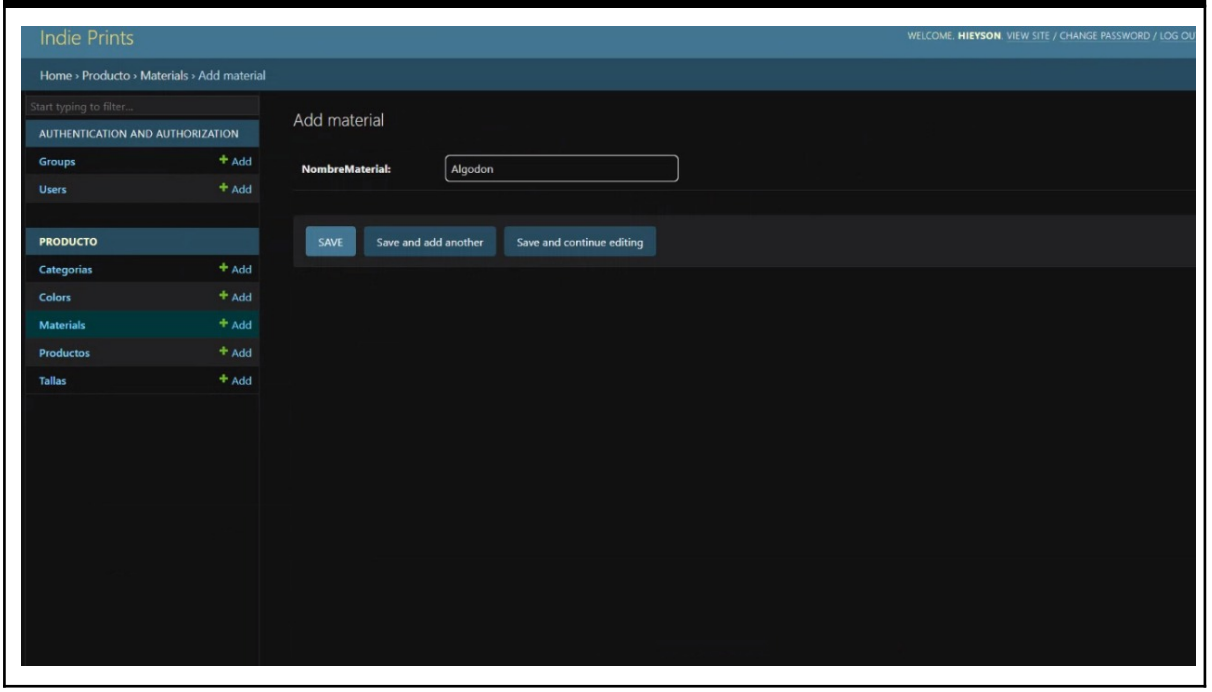


Resultado Esperado normal

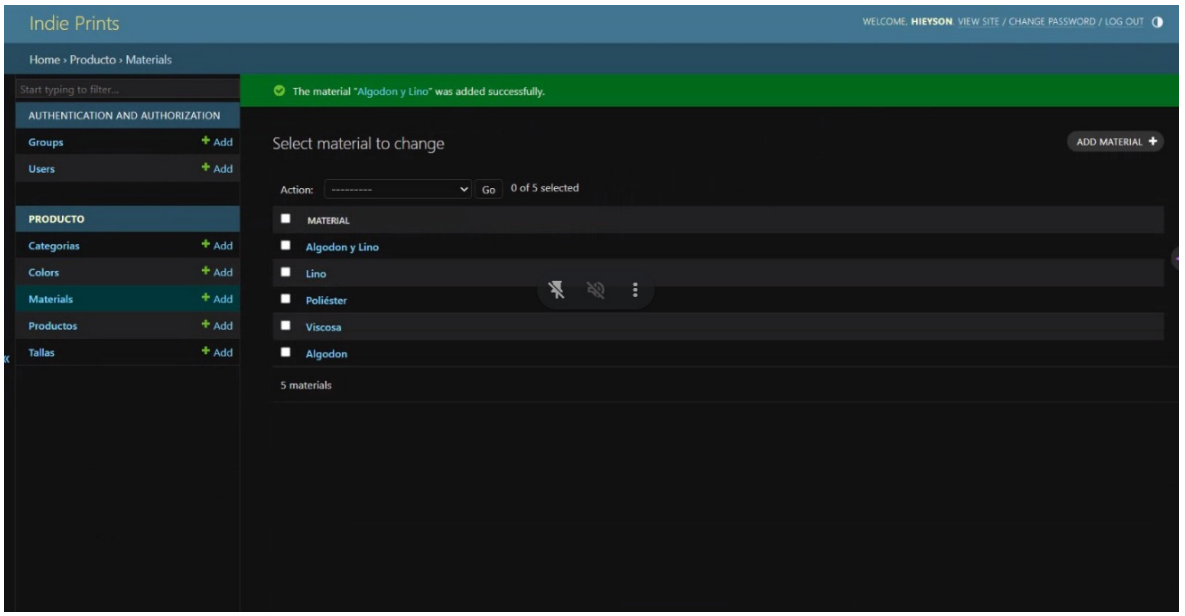


Caso de uso : Material

Datos de entrada Válidos

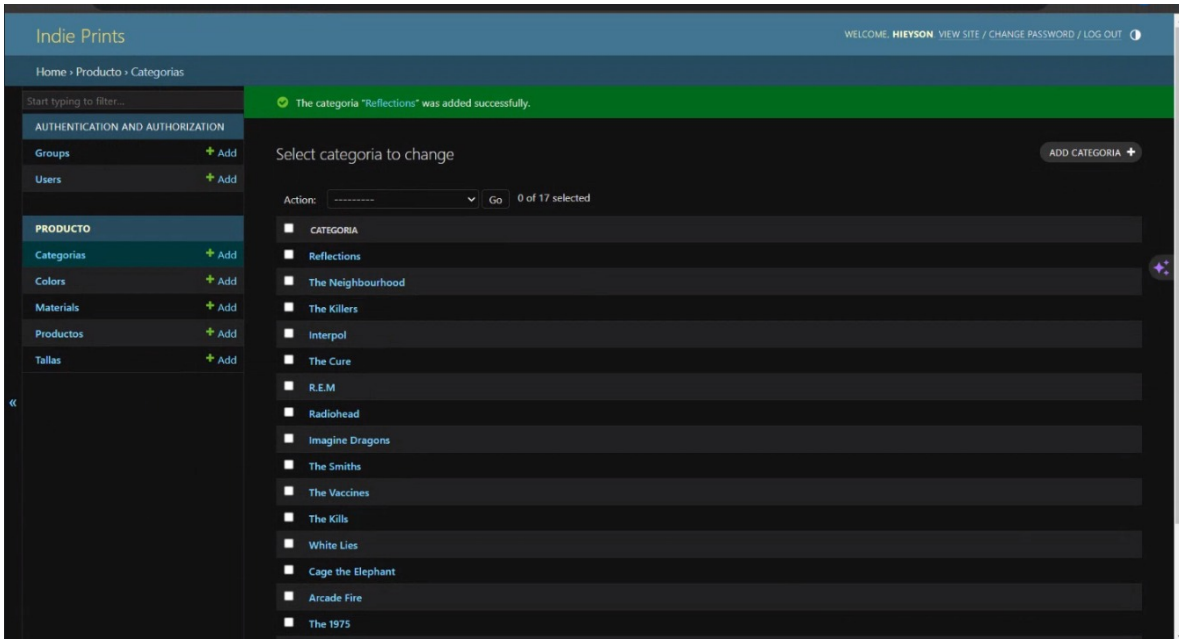


Resultado Esperado normal

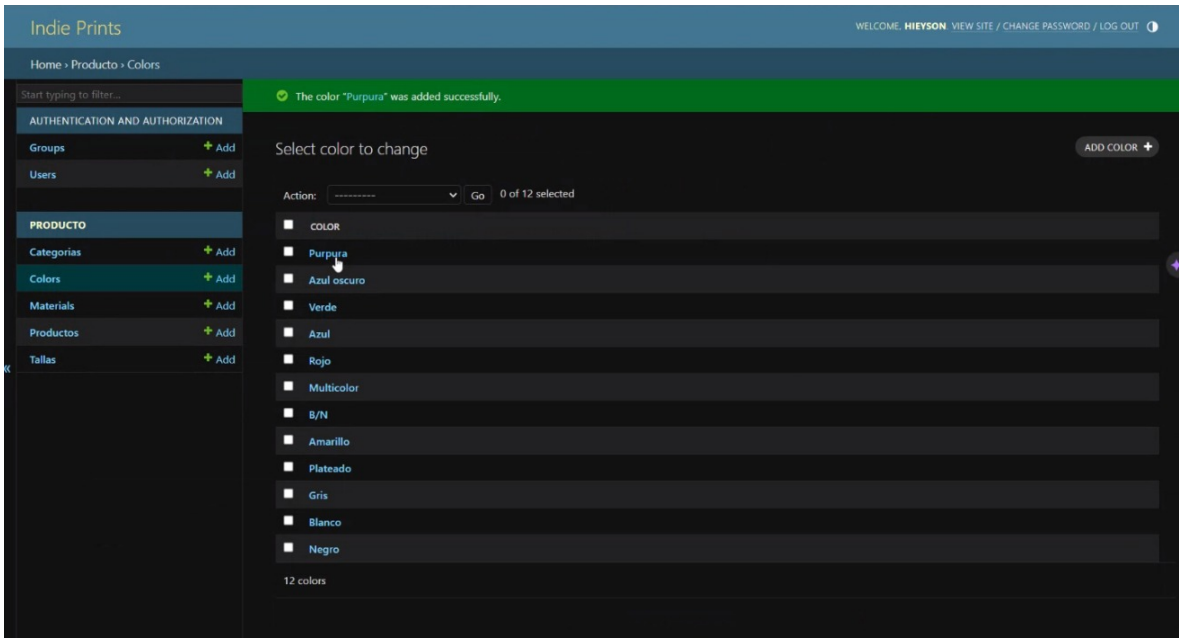


Caso de uso : Color

Datos de entrada Válidos

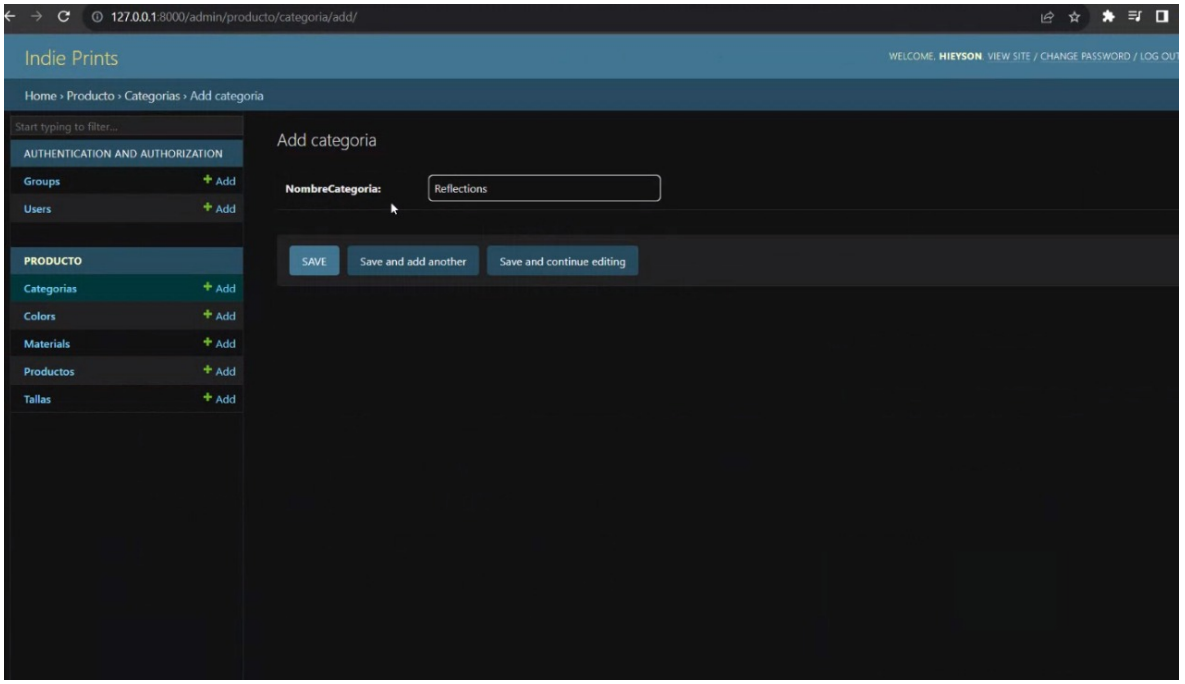


Resultado Esperado normal

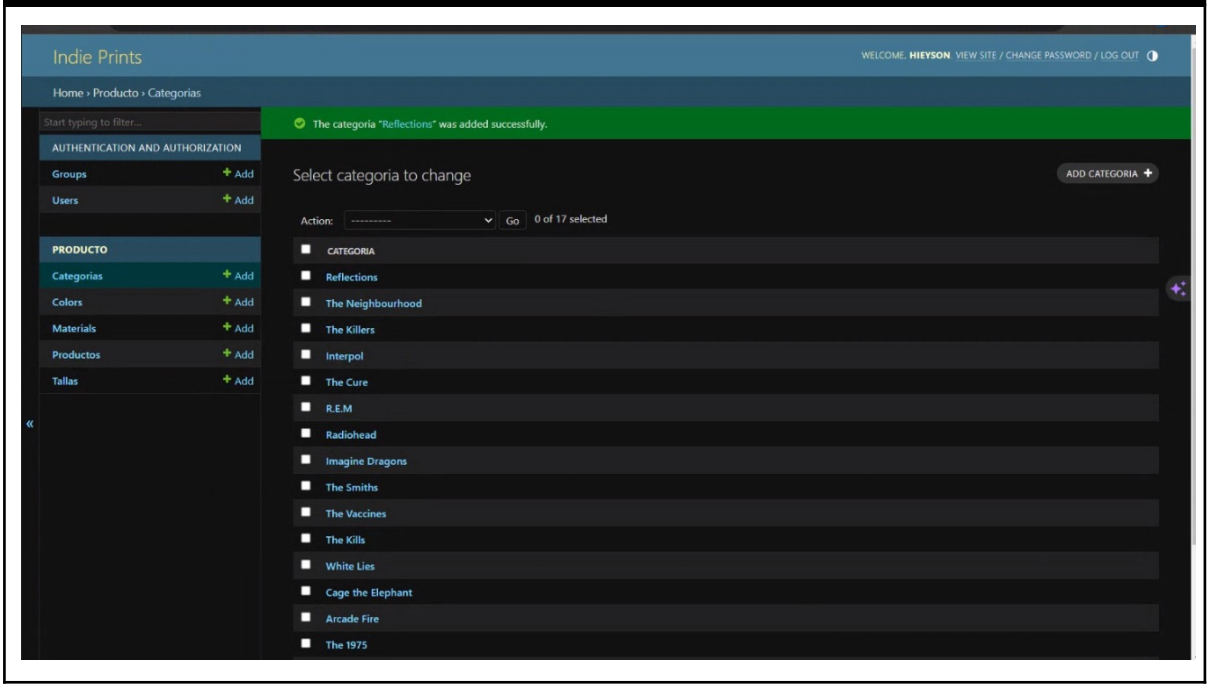


Caso de uso : Categoría

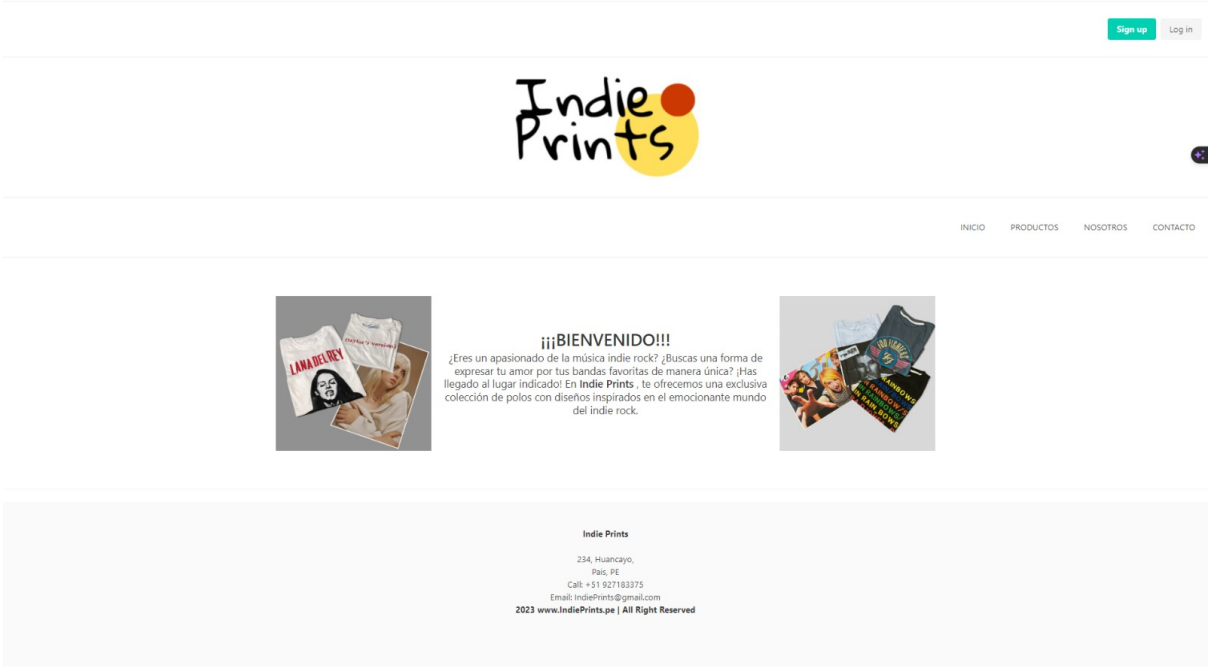
Datos de entrada Válidos



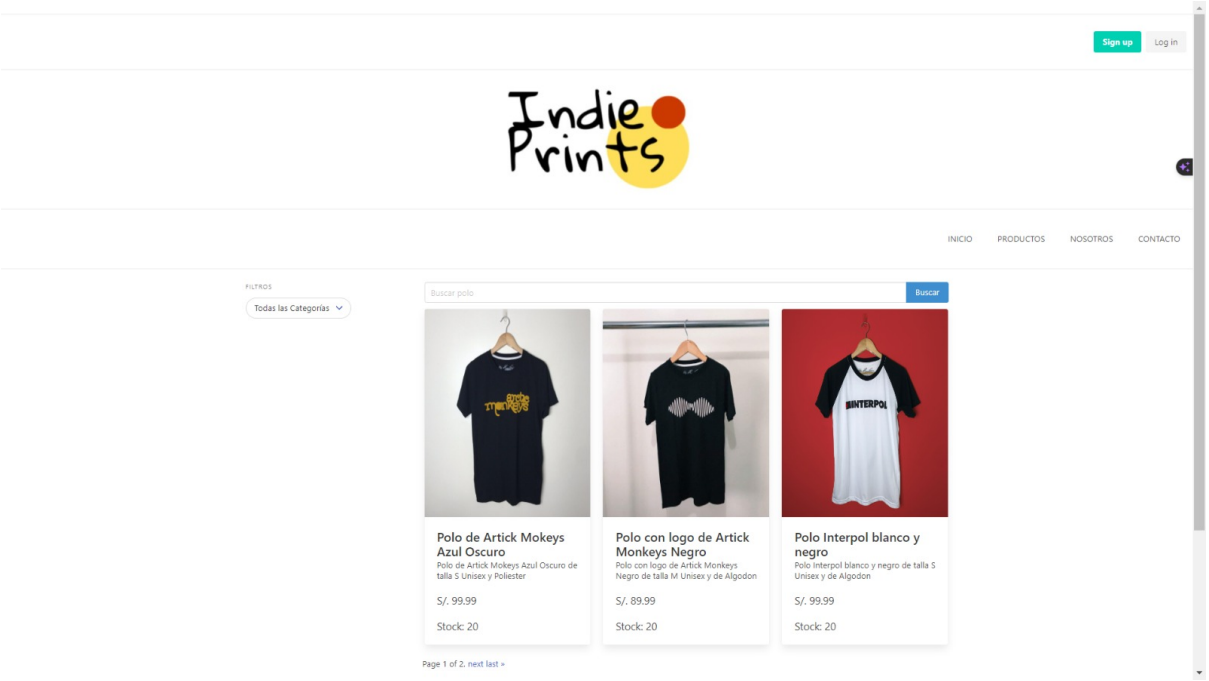
Resultado Esperado normal



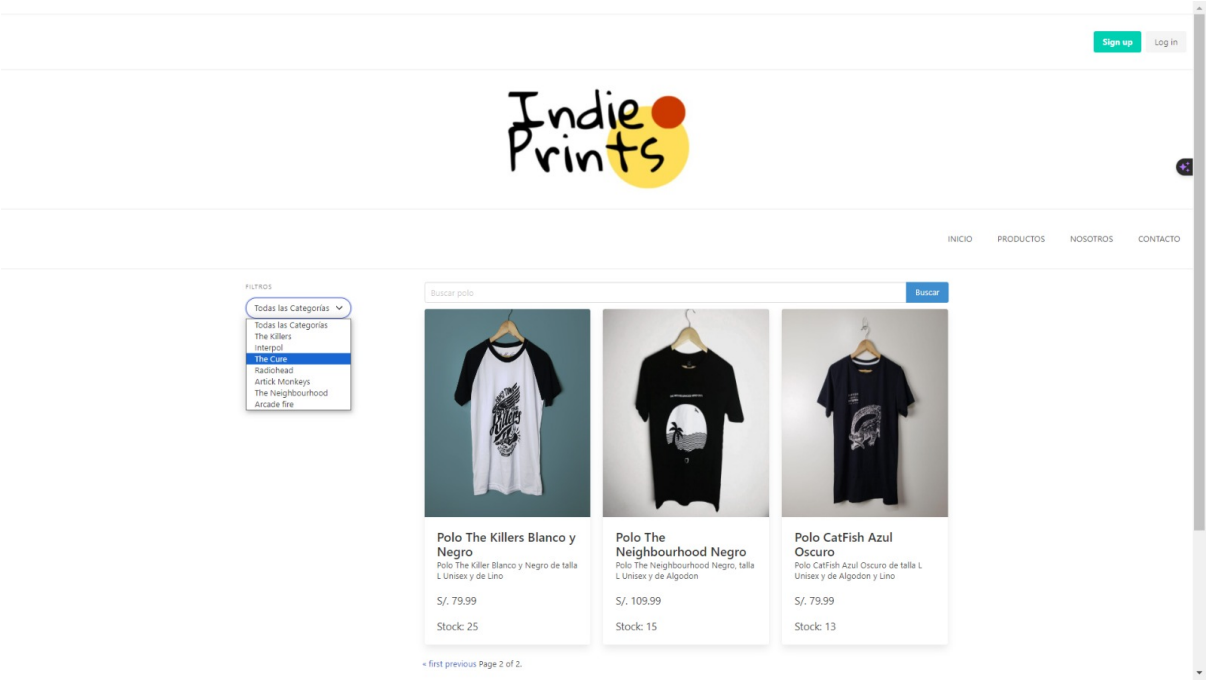
Inicio



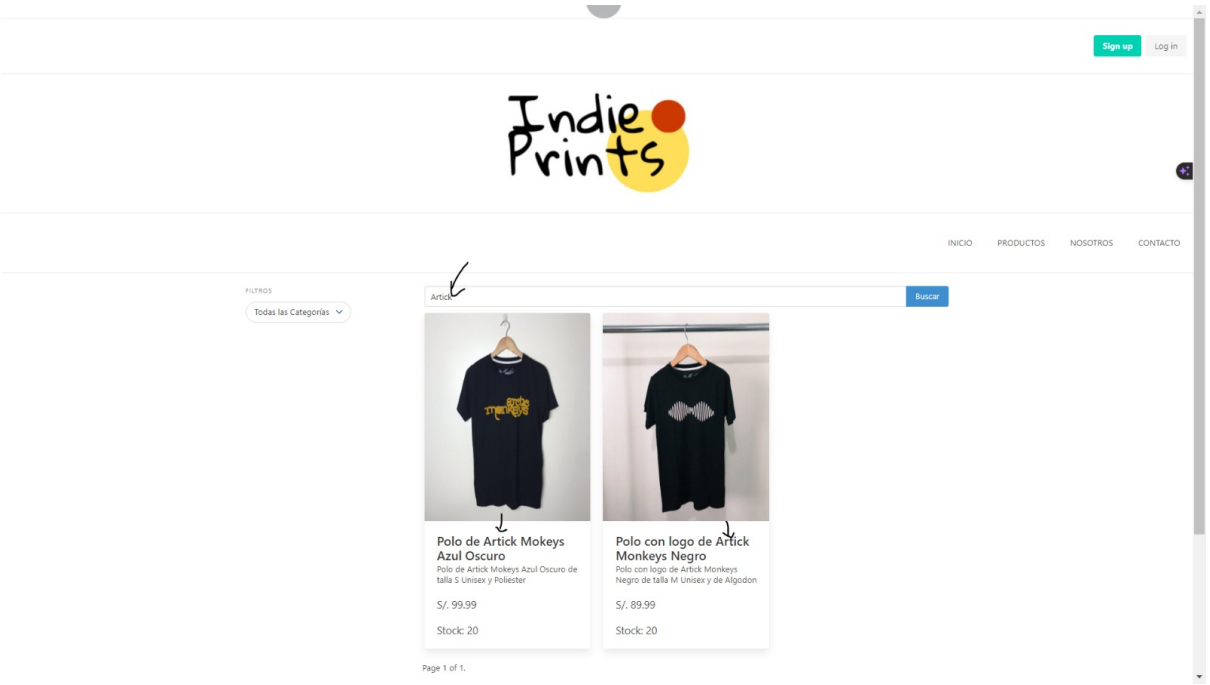
Mostrar Productos



Filtro de búsqueda por categoría



Búsqueda por nombre



Anexo:
Link Github: https://github.com/HieysonS/indiePrintsPE.git

Files

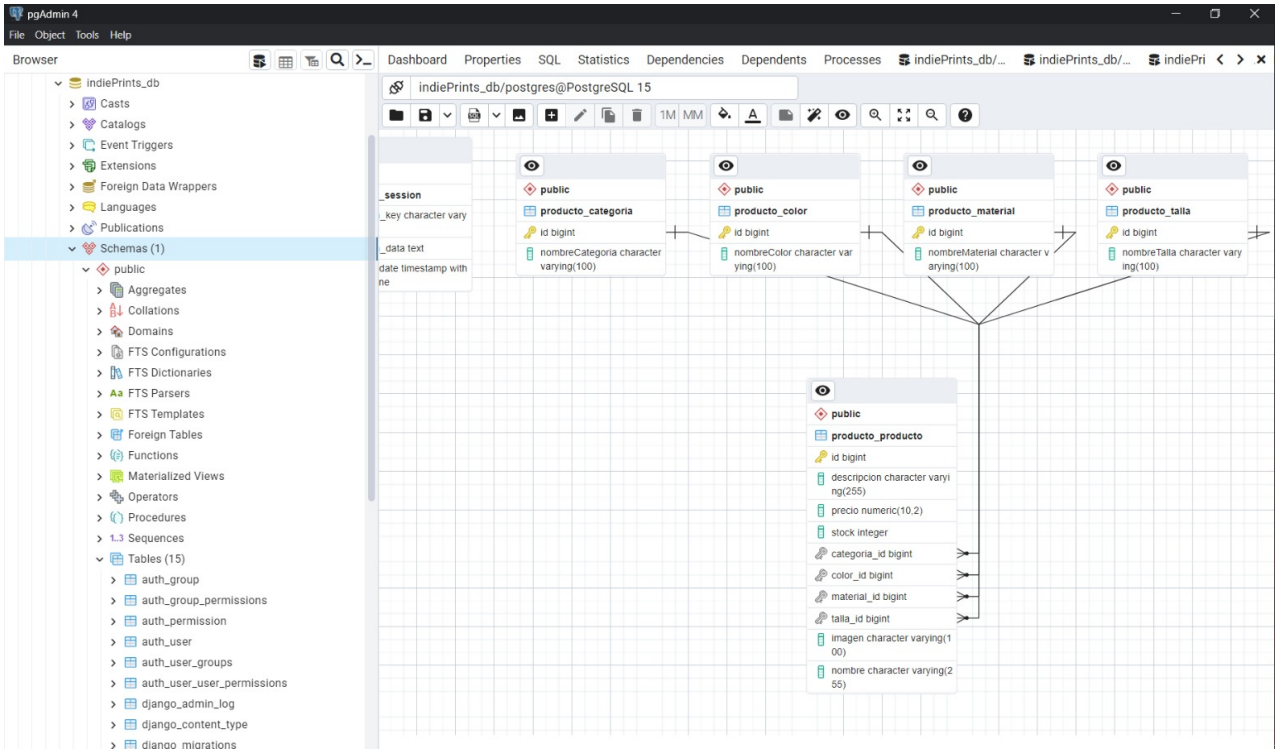
master

Go to file

- imagenes
- indiePrints
- producto
  - __pycache__
  - migrations
  - templates
  - __init__.py
  - admin.py
  - apps.py
  - models.py
  - tests.py
  - views.py
- webapp
  - __pycache__
  - migrations
  - templates
    - paginas
      - contacto.html
      - inicio.html
      - nosotros.html
    - base.html
  - __init__.py
  - admin.py

Documentation • Share feedback

indiePrintsPE / producto / models.py

Code Blame 53 lines (35 loc) · 1.6 KB

```python
class Material(models.Model):

    def __str__(self):
        return f'{self.nombreMaterial}'


class Color(models.Model):
    nombreColor = models.CharField(max_length=100, null=False)

    def __str__(self):
        return f'{self.nombreColor}'


class Categoria(models.Model):
    nombreCategoria = models.CharField(max_length=100, null=False)

    def __str__(self):
        return f'{self.nombreCategoria}'


class Producto(models.Model):
    nombre = models.CharField(max_length=255, null=False)
    talla = models.ForeignKey(Talla, on_delete=models.CASCADE)
    material = models.ForeignKey(Material, on_delete=models.CASCADE)
    color = models.ForeignKey(Color, on_delete=models.CASCADE)
    categoria = models.ForeignKey(Categoria, on_delete=models.CASCADE)
    imagen = models.ImageField(upload_to="imagenes/", verbose_name='Imagen', null=True)
    descripcion = models.CharField(max_length=255, null=False)
    precio = models.DecimalField(max_digits=10, decimal_places=2, null=False)
    stock = models.PositiveIntegerField(null=False)

    def __str__(self):
        return f'{self.nombre}' \
               f'\nPrecio: {self.precio}' \
               f'¡Aprovecha que solo queda {self.stock} unidades en stock...!'

    def delete(self, using=None, keep_parents=False):
        self.imagen.storage.delete(self.imagen.name)
        super().delete()
```