INDITDE

## Ingeniería Web

### Marcos Barcina, Ibai Guillén, Jorge El Busto, Danel Rey

[**Ingeniería Web**](#_dqiwnjki7v57) **1**

[Marcos Barcina, Ibai Guillén, Jorge El Busto, Danel Rey](#_vftvqf4a1bib) 1

[**Introducción**](#_yr5trtjvp69n) **3**

[**Parte Django / Python:**](#_epuj3dpmbzdv) **3**

[**Modelos:**](#_wlxbfuqmz19j) **3**

[**Parte JavaScript**](#_kcqlxzp2a3yn) **5**

## Introducción

En su concepción el proyecto IndiTDE implementa un sitio web en el que se muestran diferentes ofertas de ropa de todas las marcas propiedad de la empresa, clasificados por marcas (P&B, Bershka, Zara, etc.) y por categorías (Sudaderas, Chaquetas, Camisetas, etc.), pudiendo una misma prenda estar en varias categorías. También se podrán ver los detalles de cada marca y de cada categoría, incluida la lista de ofertas de ropa asociadas en cada caso.

Más adelante nuevos detalles fueron siendo decididos, como la posibilidad de clasificar las ropas por género o enviar dar la posibilidad a los clientes de enviar sugerencias.

## Parte Django / Python:

En esta sección se crean los modelos en los que se basa la aplicación, además de los formularios que requieren interacción entre el usuario y la página web y la base de datos de esta.

### Modelos:

Estas son las estructuras sobre las que se basa IndiTDE, se ubican en el archivo models.py y contiene las características de cada una de estas estructuras. El documento comienza importando elementos de django, la base de datos, los formularios y las estructuras de usuarios.

|  |
| --- |
| from django.db import models from django import forms from django.contrib.auth.models import User |

Luego se incluyen las opciones de selección que son una forma de filtrado entre elementos que IndiTDE usa para separar los productos por género.

|  |
| --- |
| TEME\_CHOICES = (('masculino', 'Masculino'), ('femenino', 'FEMENINO'), ('unisex', 'UNISEX'), ) |

Finalmente la parte principal de models.py, los modelos y sus características que los definen:

Primero tenemos la clase Carro que se usa para gestionar el array de ropa en proceso de ser comprada por cada usuario, al ser así este está simplemente compuesto del usuario al que pertenece y las diferentes prendas de ropa que contiene.

|  |
| --- |
| class Carro(models.Model):  usuario = models.ForeignKey(User, on\_delete=models.CASCADE)  ropa = models.ForeignKey('Ropa', on\_delete=models.CASCADE) |

Seguido tendríamos la marca, otra característica por la que la ropa puede ser filtrada, al ser una clase de menor importancia carece de métodos a excepción de su propio constructor, para que los artículos correspondientes puedan heredar la marca concreta.

|  |
| --- |
| class Marca(models.Model):  id = models.CharField(max\_length = 5, primary\_key =True)  nombre = models.CharField(max\_length = 20)  desc = models.TextField()  logo = models.ImageField()    def \_\_str\_\_(self):  return self.nombre |

El archivo continúa con la clase Ropa,

## Parte JavaScript