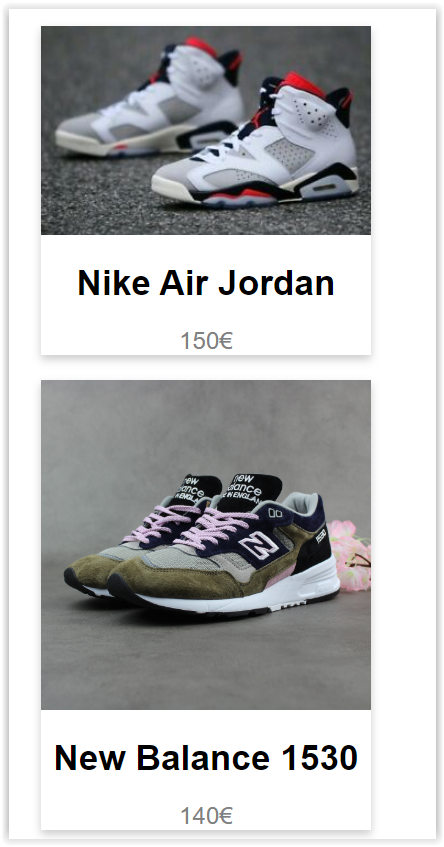
**MODELOS EN ANGULAR**

* Vamos a crear objetos de una determinada forma dentro de angular utilizando Json.
* Para ello, nos vendría bien tener un modelo para saber el tipo de dato concreto de cada propiedad a dibujar y de cada característica, ya que Type Script sí utiliza tipados y por eso es recomendable utilizar modelos.
* Un modelo simplemente es una declaración de propiedades y atributos de un

objeto.

* Para nuestro ejemplo, nos crearemos un modelo para dibujar Productos en nuestra Web.



Comenzamos creando una carpeta llamada **models** dentro de **src/app**

Dentro de la carpeta, creamos un nuevo fichero llamado **producto.ts**

**PRODUCTO.TS**

export class Producto {

  public nombre: string;

  public imagen: string;

  public precio: number;

  constructor(nombre, imagen, precio) {

    this.nombre = nombre;

    this.imagen = imagen;

    this.precio = precio;

  }

}

Creamos un nuevo **component** llamado **Productos**, dónde utilizaremos nuestro modelo recién creado.

**PRODUCTOS.COMPONENT.TS**

import { Component, OnInit } from "@angular/core";

import { Producto } from "./../../models/producto";

@Component({

  selector: "app-productos",

  templateUrl: "./productos.component.html",

  styleUrls: ["./productos.component.css"]

})

export class ProductosComponent implements OnInit {

  public productos: Array<Producto>;

  constructor() {

    //CREAMOS OBJETOS DE TIPO PRODUCTO

    this.productos = [

      new Producto(

        "Nike Air Jordan",

        "https://i.ebayimg.com/images/g/lEsAAOSwBJpc1MAN/s-l300.jpg",

        150

      ),

      new Producto(

        "New Balance 1530",

        "https://cdn.webshopapp.com/shops/164432/files/292835943/new-balance-m1530kgl-olive-grey-pink-made-in-uk.jpg",

        140

      ),

      new Producto(

        "J-Hayber Olimpo",

        "https://jhayber.com/documents/images/products/63638-850-1.jpg",

        60

      ),

      new Producto(

        "Vans",

        "https://media.vogue.es/photos/5cc75028fdc822a6611ff5bb/master/w\_2700,c\_limit/voguees\_moda\_tendencias\_250218163.jpg",

        75

      )

    ];

  }

  ngOnInit(): void {}

}

Y dibujamos los productos en la vista con algún diseño css.

**PRODUCTOS.COMPONENT.CSS**

.card {

  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);

  max-width: 300px;

  margin: auto;

  text-align: center;

  font-family: arial;

}

.price {

  color: grey;

  font-size: 22px;

}

.card button {

  border: none;

  outline: 0;

  padding: 12px;

  color: white;

  background-color: #000;

  text-align: center;

  cursor: pointer;

  width: 100%;

  font-size: 18px;

}

.card button:hover {

  opacity: 0.7;

}

**PRODUCTOS.COMPONENT.HTML**

<h1>Modelo de productos</h1>

<div class="card" \*ngFor="let producto of productos">

  <img

    src="{{ producto.imagen }}"

    alt="{{ producto.nombre }}"

    style="width:100%"

  />

  <h1>{{ producto.nombre }}</h1>

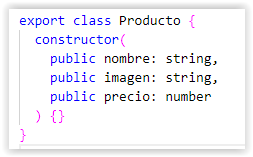
  <p class="price">{{ producto.precio }}€</p>

</div>

También podemos escribir la clase **Model** para ahorrar código en las propiedades y el constructor.

Si nos definimos las propiedades dentro del constructor, la clase seguirá siendo funcional y nos ahorraremos código en las declaraciones.

**PRODUCTO.TS**



export class Producto {

  constructor(

    public nombre: string,

    public imagen: string,

    public precio: number

  ) {}

}