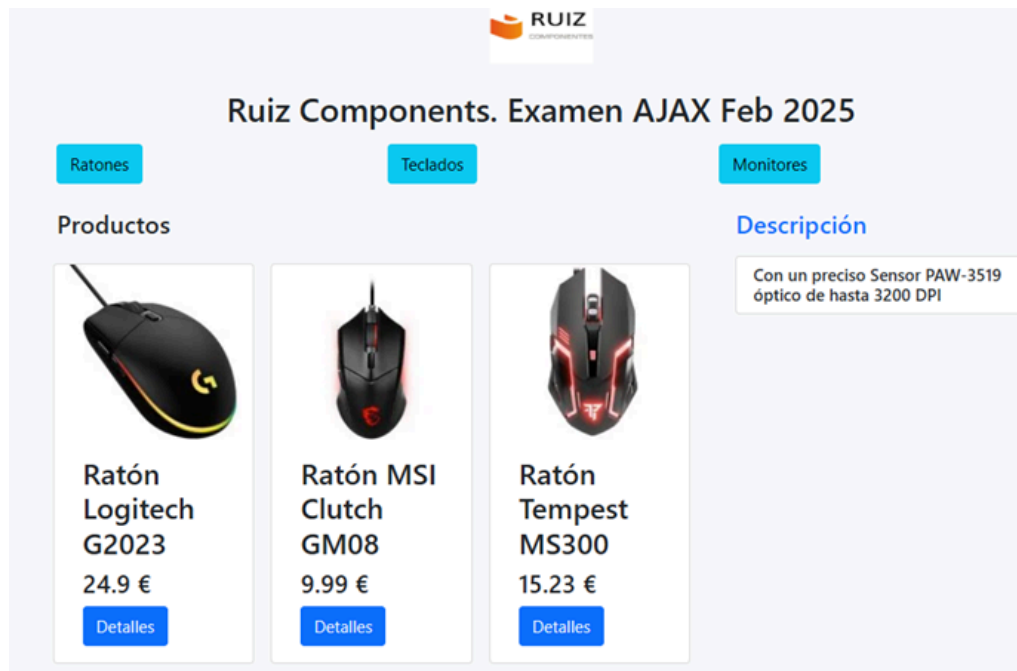


# Tienda de informática usando AJAX

<b>1. Importar la base de datos</b>	<b>2</b>
<b>2. Creación del API XML con las categorías</b>	<b>3</b>
<b>3. Consumir el API XML para crear los botones menú con las categorías</b>	<b>4</b>
<b>4. Creación del API JSON con los productos (con posibilidad de filtrado)</b>	<b>4</b>
<b>5. Consumir el API JSON para mostrar los productos</b>	<b>6</b>

Realizar una aplicación web que permita consultar información sobre productos de una tienda de informática almacenados en una base de datos. Se podrá filtrar la categoría de los productos pulsando en el botón oportuno (Ratones, Teclados, Monitores, etc.). Así mismo, al pulsar el botón Detalles, se mostrará la descripción del producto.




## 1. Importar la base de datos

La base de datos consta de dos tablas llamadas **categorías** y **productos** las cuales tienen la siguiente estructura:

Tabla **categorías**

Nombre	Tipo	<b>id</b> : identificador único de la categoría  <b>nombre</b> : nombre de la categoría (p.e. "Ratones")
id	int(11)	
nombre	varchar(50)	

Tabla **productos**

Nombre	Tipo	
id 	int(11)	<b>id</b> : identificador único del producto
id_categoria	int(11)	<b>id_categoria</b> : clave foránea al id de categoría
nombre	varchar(255)	<b>nombre</b> : nombre del producto (p.e. "Ratón Logitech G2023")
precio	decimal(10,2)	<b>precio</b> : precio del producto
descripcion	text	<b>descripcion</b> : descripción del producto
imagen	varchar(255)	<b>imagen</b> : nombre del archivo con la imagen del producto

- a) **(10 puntos)** Importa el fichero .sql suministrado por el profesor para obtener las tablas anteriores así como los registros.

## 2. Creación del API XML con las categorías

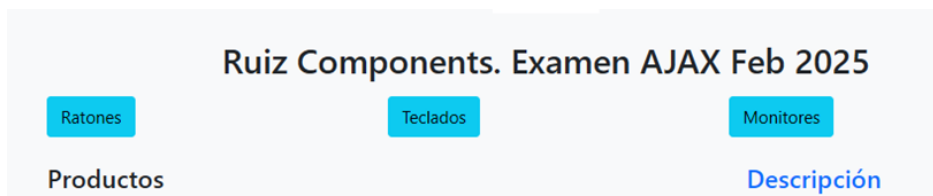
- a) Crea una API llamada **cargar\_categorias\_xml.php** que devuelva la información de las categorías en formato XML. Con los datos actuales de la BBDD la salida debería ser:

[http://localhost/examenajax/cargar\\_categorias\\_xml.php](http://localhost/examenajax/cargar_categorias_xml.php)

```
<?xml version="1.0" encoding="UTF-8"?>
<categorias>
  <categoria>
    <id>1</id>
    <nombre>Ratones</nombre>
  </categoria>
  <categoria>
    <id>2</id>
    <nombre>Teclados</nombre>
  </categoria>
  <categoria>
    <id>3</id>
    <nombre>Monitores</nombre>
  </categoria>
</categorias>
```

### 3. Consumir el API XML para crear los botones menú con las categorías

a) Realiza una petición AJAX al API XML de las categorías para crear los botones que permitirán posteriormente mostrar los productos de una determinada categoría. Los botones deberán crearse en la carga de la página web.



### 4. Creación del API JSON con los productos (con posibilidad de filtrado)

a) En este apartado tendrás que crear otro API llamada `cargar_productos_json.php` que devuelva un JSON con los productos. Tendrá un parámetro opcional llamado `id_categoria`, para indicar la categoría a filtrar. Si el API no recibe ningún parámetro devolverá todos los productos.

Por ejemplo, si llamamos al API con el parámetro `id_categoria` y valor 3 devolverá los productos con dicha categoría:

[http://localhost/examenajax/cargar\\_productos\\_json.php?id\\_categoria=3](http://localhost/examenajax/cargar_productos_json.php?id_categoria=3)

```
[
  {
    "id": 8,
    "id_categoria": 3,
    "nombre": "Monitor LG 24GS50F",
    "precio": 150.12,
    "descripcion": "Monitor diseñado especialmente para gamers",
    "imagen": "monitor_lg.webp",
    "caracteristicas": "[{"atributo": "Tipo HD", "valor": "Full HD"}, {"atributo": "Pantalla táctil", "va":
  },
  {
    "id": 9,
    "id_categoria": 3,
    "nombre": "Monitor MSG G27",
    "precio": 169.55,
    "descripcion": "Equipado con un panel de 1920x1080, 250hz",
    "imagen": "monitor_msi.webp",
    "caracteristicas": "[{"atributo": "Curvatura", "valor": "1500R"}, {"atributo": "Relación de aspecto",
      "valor": "16:9"}]"
  }
]
```

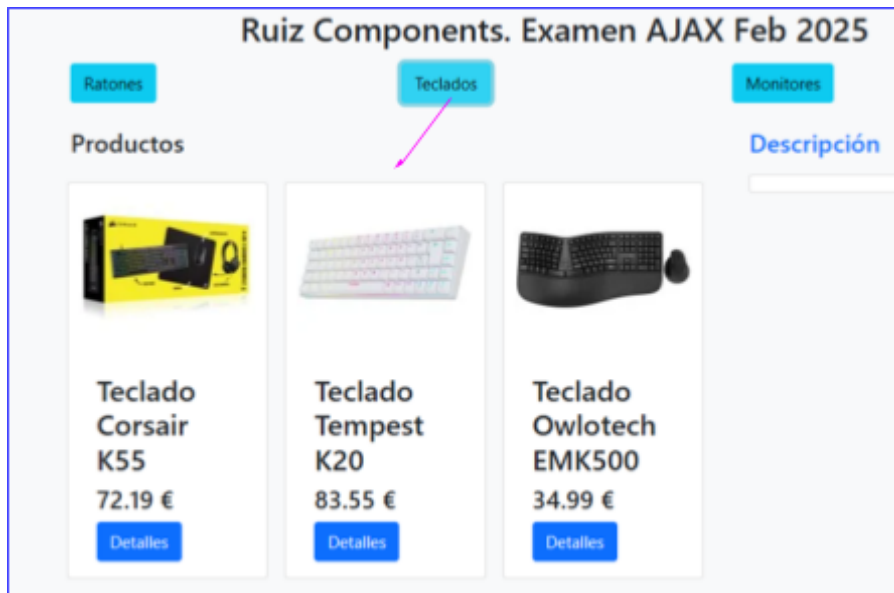
Sin invocamos al API sin el parámetro no filtrará y devolverá todos los productos:

[http://localhost/examenajax/cargar\\_productos\\_json.php](http://localhost/examenajax/cargar_productos_json.php)

```
[
  {
    "id": 1,
    "id_categoria": 1,
    "nombre": "Ratón Logitech G2023",
    "precio": 24.9,
    "descripcion": "Tecnología LIGHTSYNC, un sensor para gaming y un diseño clásico con 6 botones",
    "imagen": "logitech-g203.webp",
    "características": "[{\\"atributo\\": \\"Ancho\\", \\"valor\\": \\"34mm\\"}, {\\"atributo\\": \\"Profundidad\\", \\"valor\\": \\"{\\"atributo\\": \\"Alto\\", \\"valor\\": \\"42mm\\"}, {\\"atributo\\": \\"Peso\\", \\"valor\\": \\"85g\\"}]}]"
  },
  {
    "id": 2,
    "id_categoria": 1,
    "nombre": "Ratón MSI Clutch GM08",
    "precio": 9.99,
    "descripcion": "Con un preciso Sensor PAW-3519 óptico de hasta 3200 DPI",
    "imagen": "msi-gm08.webp",
    "características": "[{\\"atributo\\": \\"Ancho\\", \\"valor\\": \\"40mm\\"}, {\\"atributo\\": \\"Profundidad\\", \\"valor\\": \\"{\\"atributo\\": \\"Alto\\", \\"valor\\": \\"40mm\\"}, {\\"atributo\\": \\"Peso\\", \\"valor\\": \\"92g\\"}]}]"
  },
  {
    "id": 3,
    "id_categoria": 1,
    "nombre": "Ratón Tempest MS300",
    "precio": 15.23,
    "descripcion": "Ratón gaming diseñado para ofrecer precisión y estilo a los gamers más exigentes",
    "imagen": "tempest-ms300.webp",
    "características": "[{\\"atributo\\": \\"Ancho\\", \\"valor\\": \\"41mm\\"}, {\\"atributo\\": \\"Profundidad\\", \\"valor\\": \\"{\\"atributo\\": \\"Alto\\", \\"valor\\": \\"38mm\\"}, {\\"atributo\\": \\"Peso\\", \\"valor\\": \\"75g\\"}]}]"
  }
]
```

## 5. Consumir el API JSON para mostrar los productos

a) En este apartado tendrás que realizar una petición AJAX al API JSON de los productos para mostrarlo por pantalla. La petición AJAX se producirá cuando el usuario pulse alguno de los botones de las categorías.



b) En este apartado tendrás que mostrar la descripción del producto cuando el usuario pulsa el botón Detalles.

