DESARROLLO DE APLICACIONES WEB

17-11-2023

ROBERTO RODRÍGUEZ JIMÉNEZ

roberto.rodjim.1@educa.jcyl.es

Desarrollo Web en Entorno Servidor

Trabajar con bases de datos en PHP

Tarea 03

# 

Contenido

[Tarea online DWES03 2](#_Toc151122954)

[1.- Descripción de la tarea 2](#_Toc151122955)

[**Caso práctico** 2](file:///H:\CFGS\daw\dwes\rodriguez_jimenez_roberto_DWES03_Tarea\rodriguez_jimenez_roberto_DWES03_Tarea.docx#_Toc151122956)

[Enunciado 2](#_Toc151122957)

[2.- Información de interés 5](#_Toc151122958)

[Recursos necesarios 5](#_Toc151122959)

[Consejos y recomendaciones 5](#_Toc151122960)

[3.- Evaluación de la tarea 5](#_Toc151122961)

[Criterios de evaluación implicados 5](#_Toc151122962)

[¿Cómo valoramos y puntuamos tu tarea? 6](#_Toc151122963)

[RESOLUCIÓN 7](#_Toc151122964)

# Tarea online DWES03

**Título de la tarea**: Cesta de la compra.  
**Unidad 3**: Trabajar con bases de datos en PHP.  
**Ciclo formativo y módulo**: Desarrollo de Aplicaciones Web. Módulo DWES.  
**Curso académico**: …

## 1.- Descripción de la tarea

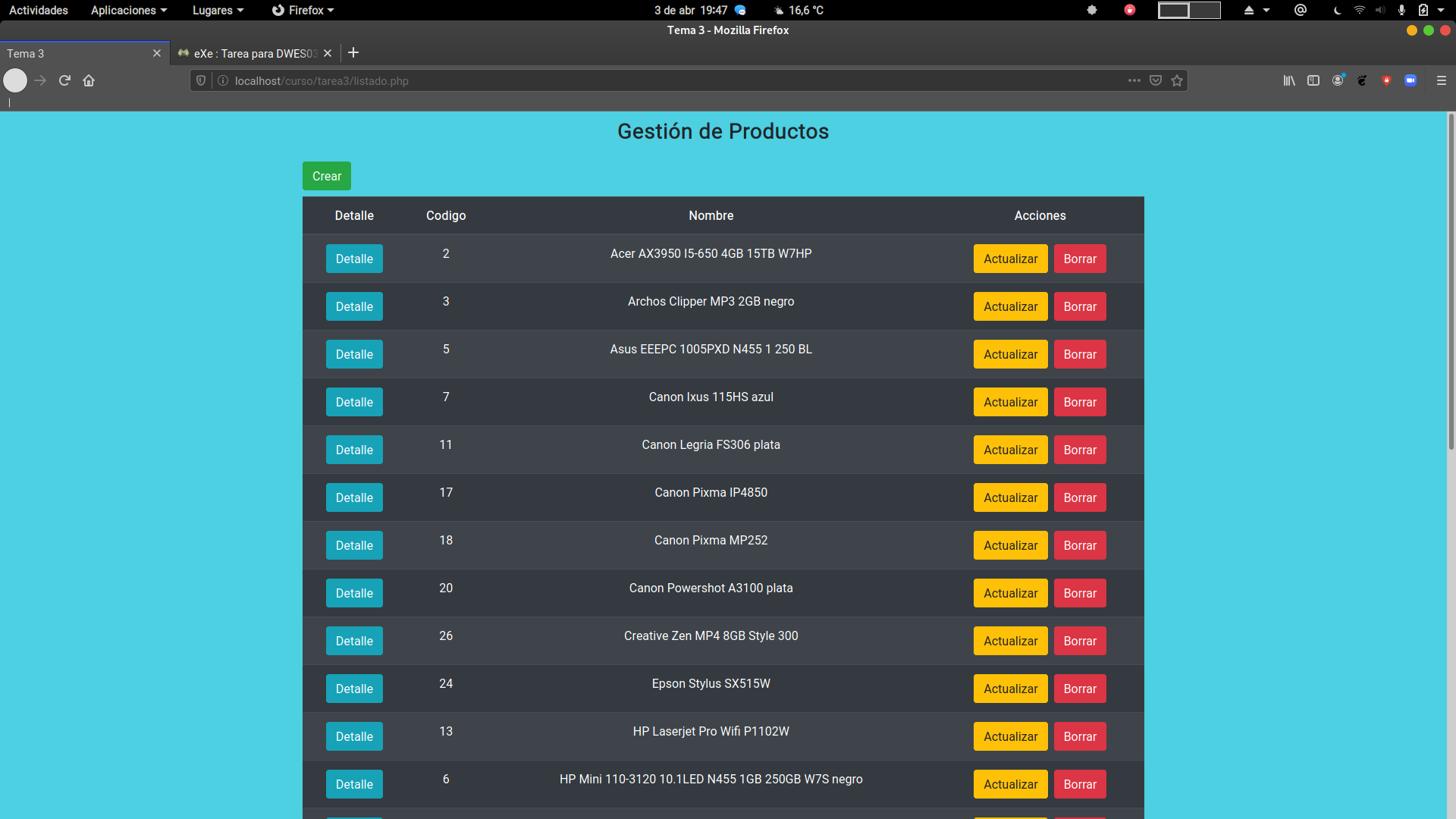
### **Caso práctico**

Carlos tiene un amigo con una tienda de informática, que ha decidido vender sus productos por Internet. Carlos piensa que con sus conocimientos de acceder a una base de datos desde la Web podría echarle una mano, y aunque es consciente de que aún le faltan herramientas para que la aplicación sea completamente funcional, se pone a hacer una primera implementación de esta.

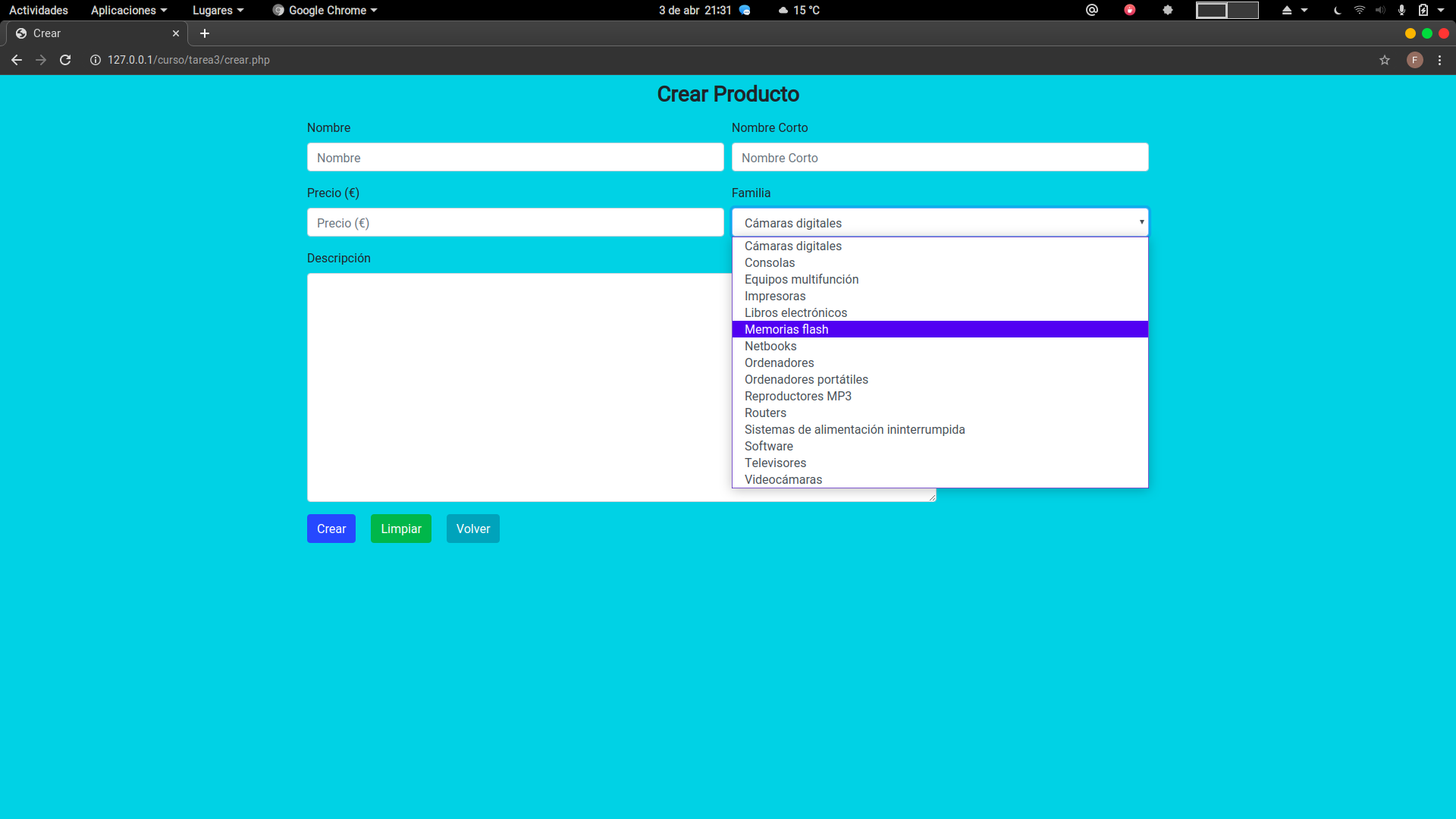
### Enunciado

Partiendo de la base de datos 'proyecto' usada en los ejemplos y ejercicios de la unidad , se trata de programar un CRUD (create, read, update, delete) que permita gestionar los registros de la tabla 'productos'. La aplicación se dividirá en 5 páginas:

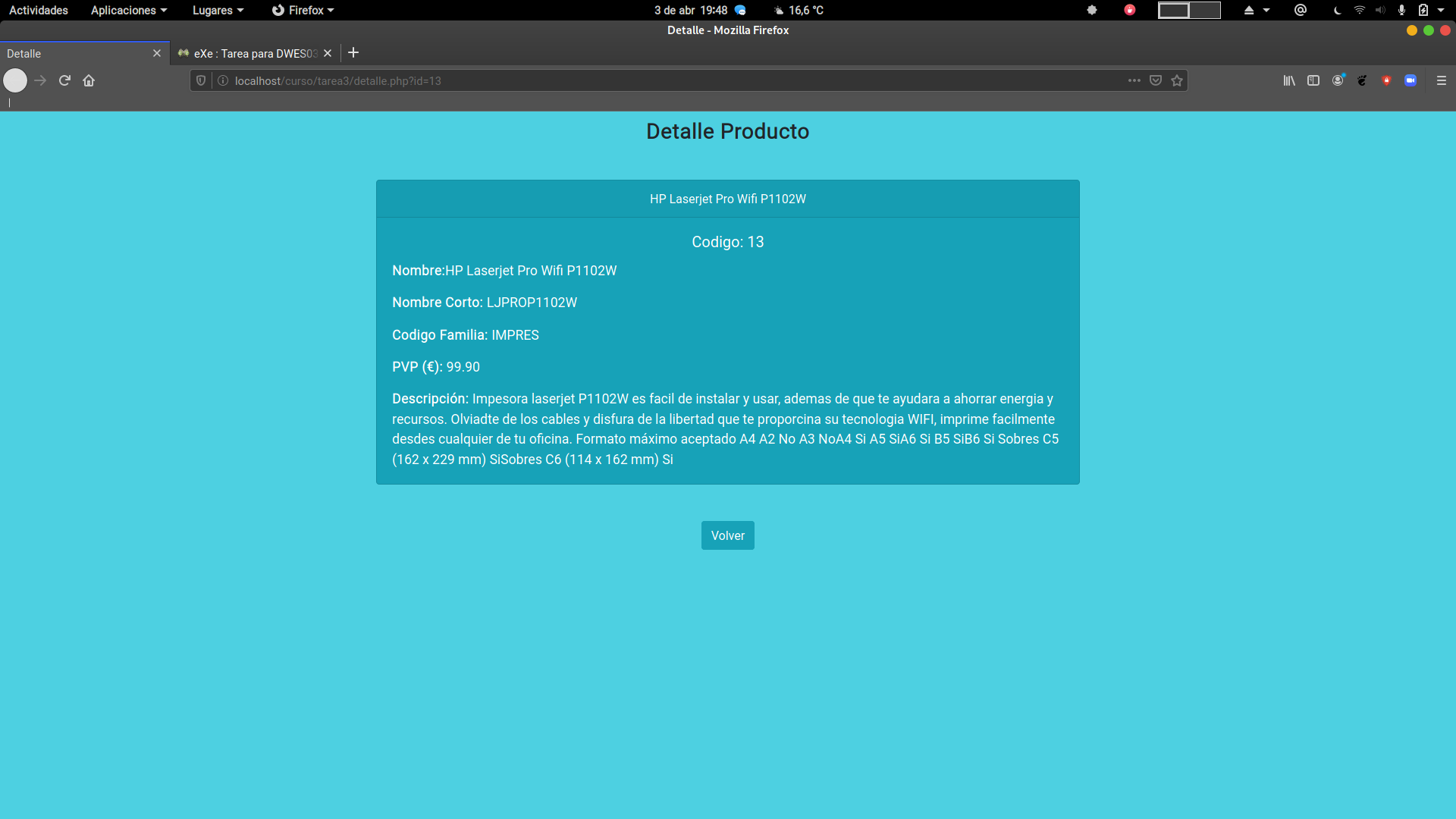
* listado.php. Mostrará en una tabla los datos código y nombre y los botones para crear un nuevo registro, actualizar uno existente, borrarlo o ver todos sus detalles.



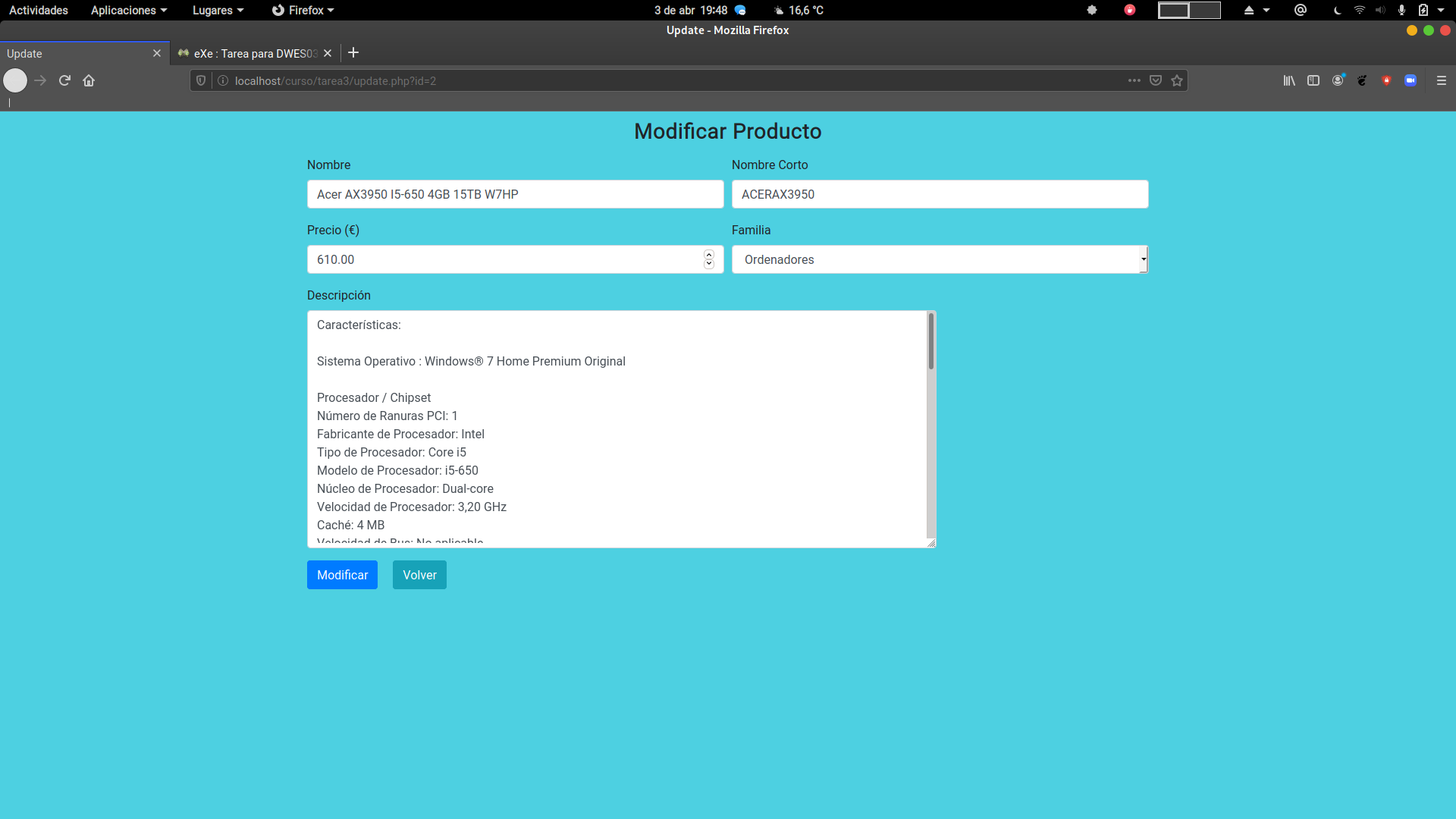
* crear.php. Será un formulario para rellenar todos los campos de productos (a excepción del id). Para la familia nos aparecerá un "select" con los nombres de las familias de los productos para elegir uno (lógicamente, aunque mostremos los nombres por formulario enviaremos el código).



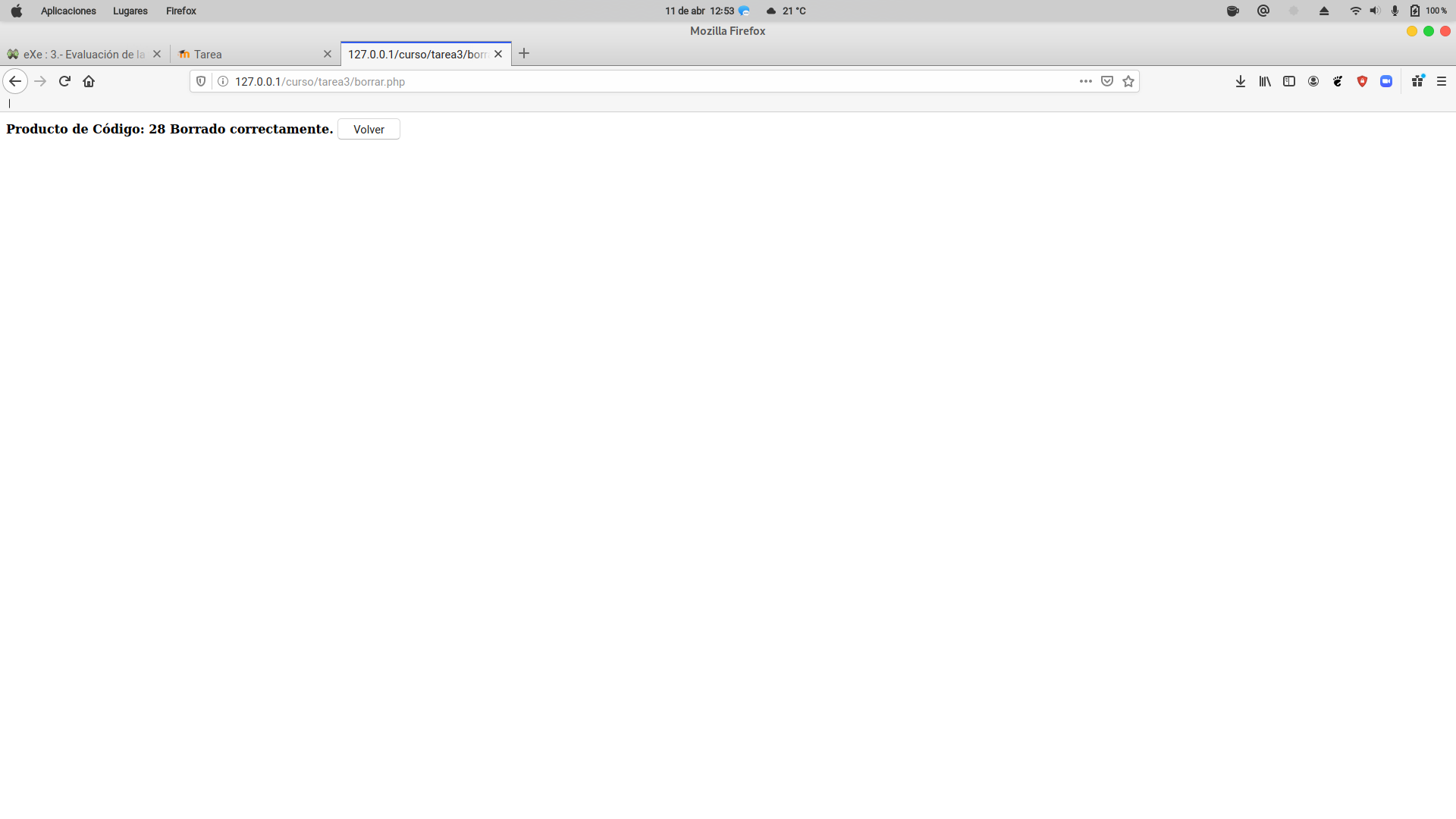
* detalle.php. Mostrará todos los detalles del producto seleccionado.



* update.php. Nos aparecerá un formulario con los campos rellenos con los valores del producto seleccionado desde "listado.php" incluido el select donde seleccionamos la familia.



* borrar.php. Será una página php con el código necesario para borrar el producto seleccionado desde "listado.php" un mensaje de información y un botón volver para volver a "listado.php".



Para acceder a la base de datos se debe usar PDO. Controlaremos y mostraremos los posibles errores. Para los estilos se recomienda usar *Bootstrap*.

Pasaremos el código de producto por "get" tanto para "detalle.php" como para "update.php". Utilizando en el enlace "detalle.php?id=cod" .En ambas páginas comprobaremos que esta variable existe, si no redireccionaremos a "listado.php" para esto podemos usar "header('Location:listado.php')"

## 2.- Información de interés

### Recursos necesarios

Ordenador con un entorno AMP, instalado y configurado, un entorno de desarrollo como VSC con las extensiones adecuadas. Navegador con acceso a Internet, para poder consultar el manual online de PHP y poder trabajar online Bootstrap.

### Consejos y recomendaciones

* Además del manual online de PHP, se recomienda dar libre acceso a Internet para la búsqueda de información.
* Para no repetir el código de la conexión a la base de datos en cada archivo, se recomienda crear el archivo conexion.php y utilizar require o require\_once cada vez que lo necesitamos.
* Para borrar un producto, crea un formulario con el action apuntando a "borrar.php" y pasa por un campo oculto el código del producto a borrar.
* Recuerda que en el "option" del HTML "<option value="v">Nombre</option>", "v" es el valor que pasamos, "Nombre" es lo que mostramos en la lista desplegable.

## 3.- Evaluación de la tarea

### Criterios de evaluación implicados

* Se han analizado las tecnologías que permiten el acceso mediante programación a la información disponible en almacenes de datos.
* Se han creado aplicaciones que establezcan conexiones con bases de datos.
* Se ha recuperado información almacenada en bases de datos.
* Se ha publicado en aplicaciones Web la información recuperada.
* Se han utilizado conjuntos de datos para almacenar la información.
* Se han creado aplicaciones Web que permitan la actualización y la eliminación de información disponible en una base de datos.
* Se han utilizado transacciones para mantener la consistencia de la información.
* Se han probado y documentado las aplicaciones.

### ¿Cómo valoramos y puntuamos tu tarea?

|  |  |
| --- | --- |
| **Rúbrica de la tarea** | |
| En "listado.php", generar el listado de los productos y los botones funcionando, el botón borrar será el submit de un formulario cuyo action debe ser el indicado | 2 puntos |
| En "crear.php", generar el formulario funcional de creación del producto | 2 puntos |
| En "update.php" cargar el formulario y que sea funcional con todos los campos del producto en cuestión rellenos. | 2 puntos |
| Página "borrar.php" funcional con mensaje y botón para volver. | 1 Punto |
| Página "detalle.php", mostrando el detalle del producto seleccionado y el botón volver funcionando | 1 Punto |
| La lista de selección **familia** de la página "update.php" tiene seleccionada la familia del producto. | 0.75 Puntos |
| Se utilizan correctamente las excepciones para controlar los posibles errores | 0.5 Puntos |
| Introducir comentarios, legibilidad del código y diseño | 0.75 Puntos |

## PREVIO

* SO: Ubuntu 22.04 (máquina virtual en Virtual Box).
* Acceso a los SGBD:
  + PDO: es válido para diferentes motores del BBDD.
    - Drivers
      * libmysql: no está optimizado para PHP.
      * mysqlnd: driver nativo de MySQL, mejorado.
  + ~~MySQLi: extensión nativa.~~
    - ~~Programación dual: POO y procedimental.~~
* SGBD:
  + ~~SQLite.~~
  + ~~MySQL~~
  + MariaDB.
    - Motor de la BD
      * ~~MyISAM: rápido pero no contempla integridad referecnial ni transacciones.~~
      * InnoDB: más lento pero soporta integridad referencial y tablas transaccionales.
        + Cada consulta se incluye en su propia transacción.
    - Lenguaje: SQL.
  + ~~Oracle.~~
  + ~~SQL Server.~~

## 

## RESOLUCIÓN