



Universidad  
Técnica de  
Manabí  
desde 1952



CIENCIAS  
INFORMÁTICAS

UTM  
online

# DESARROLLO DE APLICACIONES WEB

**FACULTAD DE CIENCIAS INFORMÁTICAS**

**CARRERA: TECNOLOGÍA DE LA INFORMACIÓN - ONLINE**



**TEMA: “Microservicios Actividad 6”**

**ESTUDIANTE: PINARGOTE CHICA LILIAN THAMARA**

**NIVEL: SEPTIMO**

**PARALELO: “A”**

***Periodo Académico  
Octubre 2023 – Enero 2024***



## Titulo.

Revise la documentación y cree una aplicación con arquitectura de microservicios aplicando los conocimientos adquiridos en clase.

## Desarrollo.

Una arquitectura de microservicios es una forma de diseñar y desarrollar sistemas de software como una colección de servicios pequeños e independientes, cada uno ejecutándose en su propio proceso y comunicándose mediante mecanismos livianos como HTTP/REST o protocolos de mensajería.

### Nuevo Archivo para el Microservicio:

Creo un nuevo archivo ***microservicio\_productos.php***. Este archivo contendrá la lógica específica para obtener los productos.

```
microservicio_productos.php > ...
1  <?php
2  // microservicio_productos.php
3  include_once "funciones.php";
4
5  function obtenerProductosParaMicroservicio() {
6      // Mantiene la lógica existente de obtenerProductos()
7      $bd = obtenerConexion();
8      $sentencia = $bd->query("SELECT id, nombre, descripcion, precio FROM productos");
9      return $sentencia->fetchAll();
10 }
11 >
12 |
```

Luego en ***productos.php***, incluyo y utilizo el microservicio.

```
productos.php > div.columns > div.column > table.table > tbody
1  <?php include_once "index.php" ?>
2  <?php
3  include_once "funciones.php";
4  include_once "microservicio_productos.php";
5
6  $productos = obtenerProductosParaMicroservicio();
7  >
8  <div class="columns">
9      <div class="column">
10         <h2 class="is-size-2">Productos existentes</h2>
11         <a class="button is-success" href="agregar_producto.php">Nuevo<i class="fa fa-plus"></i></a>
12         <table class="table">
13             <thead>
14                 <tr>
15                     <th>Nombre</th>
16                     <th>Descripción</th>
17                     <th>Precio</th>
18                     <th>Eliminar</th>
19                 </tr>
20             </thead>
21             <tbody>
22                 <?php foreach ($productos as $producto) { ?>
23                     <tr>
24                         <td><?php echo $producto->nombre ?></td>
25                         <td><?php echo $producto->descripcion ?></td>
26                         <td><?php echo number_format($producto->precio, 2) ?></td>
27                         <td>
28                             <form action="eliminar_producto.php" method="post">
29                                 <input type="hidden" name="id_producto" value="<?php echo $producto->id ?>">
30                                 <button class="button is-danger">
31                                     <i class="fa fa-trash-o"></i>
32                                 </button>
```

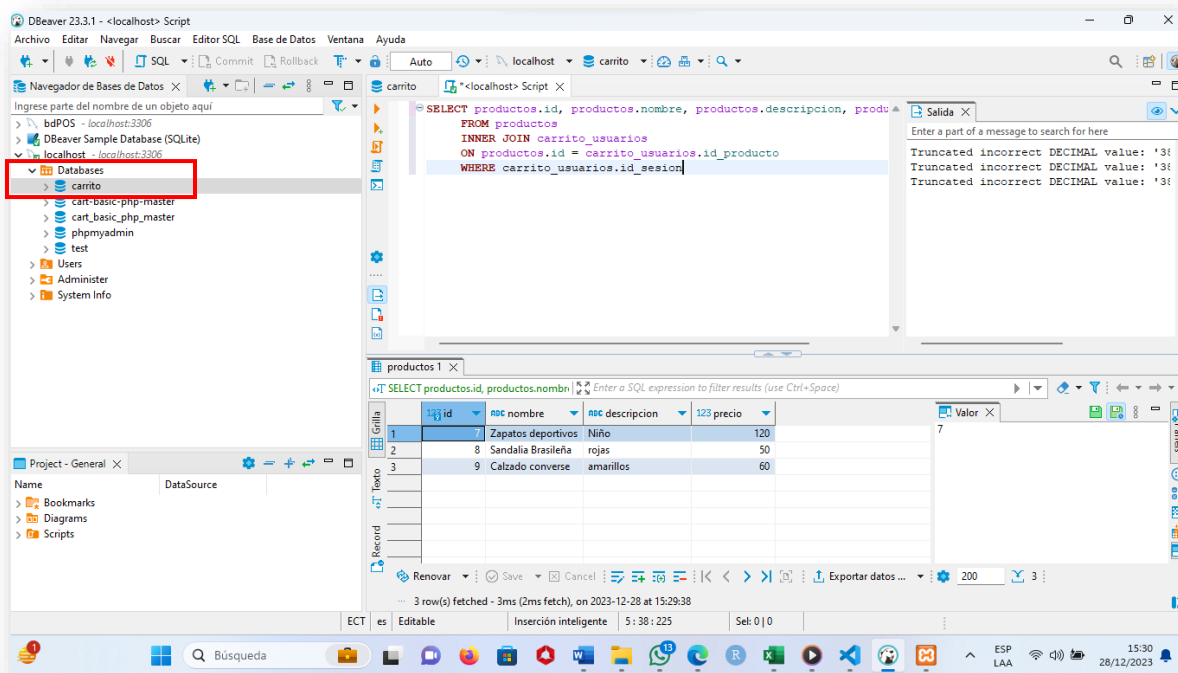


Podemos observar como obtenemos el producto.

```
function obtenerProductosEnCarrito()
{
    $bd = obtenerConexion();
    iniciarSesionSiNoEstaIniciada();
    $sentencia = $bd->prepare("SELECT productos.id, productos.nombre, productos.descripcion, productos.precio
FROM productos
INNER JOIN carrito_usuarios
ON productos.id = carrito_usuarios.id_producto
WHERE carrito_usuarios.id_sesion = ?");

    $idSesion = session_id();
    $sentencia->execute([$idSesion]);
    return $sentencia->fetchAll();
}
```

De esta forma podemos observar la tabla de nuestra base de datos en Dbeaver



The screenshot shows the DBeaver 23.3.1 interface. On the left, the 'Navegador de Bases de Datos' (Database Navigator) pane shows a tree structure with 'Databases' expanded, and 'carrito' selected. The main editor shows a SQL query:

```
SELECT productos.id, productos.nombre, productos.descripcion, productos.precio
FROM productos
INNER JOIN carrito_usuarios
ON productos.id = carrito_usuarios.id_producto
WHERE carrito_usuarios.id_sesion = ?
```

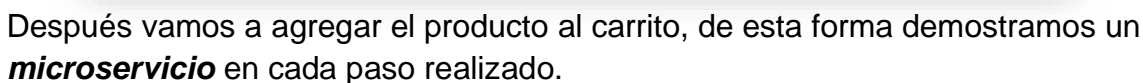
Below the query, the 'productos 1' table is displayed with the following data:

id	nombre	descripcion	precio
1	Zapatos deportivos	Niño	120
2	Sandalia Brasileña	rojas	50
3	Calzado converse	amarillos	60

The status bar at the bottom indicates '3 row(s) fetched - 3ms (2ms fetch), on 2023-12-28 at 15:29:38'.

Aquí podemos quitar el producto.

```
function quitarProductoDelCarrito($idProducto)
{
    $bd = obtenerConexion();
    iniciarSesionSiNoEstaIniciada();
    $idSesion = session_id();
    $sentencia = $bd->prepare("DELETE FROM carrito_usuarios WHERE id_sesion = ? AND id_producto = ?");
    return $sentencia->execute([$idSesion, $idProducto]);
}
```



SQL Editor: `SELECT id_producto FROM carrito_usuarios WHERE id_sesion`

Query Results (Grilla):

	id_producto
1	7
2	8
3	9

Dictionary (productos):

Value	Description
7	Zapatos deportivos
8	Sandalia Brasileña
9	Calzado converse

En conclusión, de los microservicios, el trabajo realizado está relacionado con un sistema de carrito de compras implementado en PHP y utilizando una base de datos (en este caso, MySQL/MariaDB).



Las funciones del carrito de compras están en un solo conjunto de archivos (PHP y funciones asociadas), efectúe funciones relacionadas con el carrito de compras, como agregar productos al carrito, obtener productos en el carrito, quitar productos del carrito, etc, la conexión a Base de Datos se estableció a MySQL/MariaDB para almacenar información sobre productos y el estado del carrito para usuarios, y por último Trabaje con DBeaver para ejecutar consultas SQL directamente en la base de datos.