

INTEGRACIÓN CONTINUA GRUPO B01

ENTREGA 1

Proyecto Integración Continua

Profesor:

Oscar Vargas

INTEGRANTES:

Andrés Felipe Villegas Corrales. Código: 100248416

Alexander Rodriguez Rodriguez. Código: 100234286

Yeferson Eduardo Noreña Arenas: 100093935

Sergio Ricardo Figueroa Salcedo. Código: 1921023207

Brayan Mauricio Moreno Pedraza Código: 2011024704

**INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO
COLOMBIA**

2022

TABLA DE CONTENIDO

DESARROLLO DE LA ACTIVIDAD	3
DESCRIPCIÓN DEL PROYECTO	3
LENGUAJE DE PROGRAMACIÓN Y TECNOLOGÍAS:	4
DIAGRAMA DE ARQUITECTURA GENERAL.....	4
ROLES DEL EQUIPO	5
DESARROLLO DEL PROYETO.....	6
LINK DEL VIDEO	11

DESARROLLO DE LA ACTIVIDAD

Cómo se deberá realizar la entrega de la semana 3 (una sola persona del grupo debe subir el trabajo):

- Descripción del proyecto.
- Lenguaje seleccionado.
- Diagrama de arquitectura general.
- Roles en el equipo.
- Link a video del grupo explicando: [Expo v1.mp4](#)
 - Explicación con ejemplo de uso de como realizaron la conexión de los 2 contenedores en donde se evidencie avance o como esto les ayudará en el desarrollo de su proyecto. (Video máximo de 4 minutos).
 - En el video deben aparecer todos los integrantes del grupo y deben intervenir en la explicación (Cámara encendida de todos los integrantes).
 - <https://drive.google.com/drive/folders/1DNF2CxVmxIjVBEaOwyPLvO4rE1Cz29Nf>

DESCRIPCIÓN DEL PROYECTO

Realizaremos un E-Commerce o tienda de zapatos que nos permitirá buscar diversos productos y seleccionarlos, así como poder comprarlos.

Realizaremos un servicio Web y se le creara una base de datos, en la cual podamos ingresar a la página, un servicio de registro y logueo de usuarios, en la cual podamos comprar a través de un carrito de compras ciertos productos. Este proyecto lo vamos a realizar en la plataforma Docker, a través de dos contenedores, un contenedor llamado php (apache) con nuestra página Web y el otro contenedor para nuestra imagen de SQL para la base de datos. Conectaremos los contenedores a través de una propiedad llamada --link.

Las ventajas de tener una tienda virtual en lugar de un local físico (o de tener las dos) son diversas:

Tu público potencial es mayor porque no está restringido únicamente a las personas que se encuentran cerca de una determinada ciudad o zona geográfica.

Tiene menos gastos de mantenimiento, ya que para vender por internet no se necesita invertir en alquiler, limpieza, equipamientos, decoración, etc. Ofrece más flexibilidad para el día a día, puedes estar en la playa, en la cama o en un bar y aun así trabajar en tu E-Commerce vendiendo y ganando dinero.

Es relativamente más seguro, pues cuentas con una plataforma completa, con herramientas antifraude y una empresa confiable. Por tanto, la integridad de tu negocio y de tus finanzas está garantizada.

LENGUAJE DE PROGRAMACIÓN Y TECNOLOGÍAS:

El lenguaje que vamos a utilizar es: HTML, CSS, JavaScript

Versión php Apache/2.4.54 (Debian) PHP/7.4.33 para que reconozca los comandos de php.

Dos contenedores Docker. Docker version 20.10.20, build 9fdeb9c. Para realizar la conexión entre ambos contenedores utilizamos la herramienta –Link. Los enlaces permiten que los contenedores se descubran entre sí y transfieran información de forma segura sobre un contenedor a otro contenedor. Cuando configura un enlace, crea un conducto entre un contenedor de origen y un contenedor de destino. El destinatario puede entonces acceder a datos selectos sobre la fuente. Para crear un enlace, utiliza el indicador --link. Primero, cree un nuevo contenedor, esta vez uno que contenga una base de datos.

Motor de base de datos: Server version 5.7.40 MySQL Community Server (GPL).

DIAGRAMA DE ARQUITECTURA GENERAL

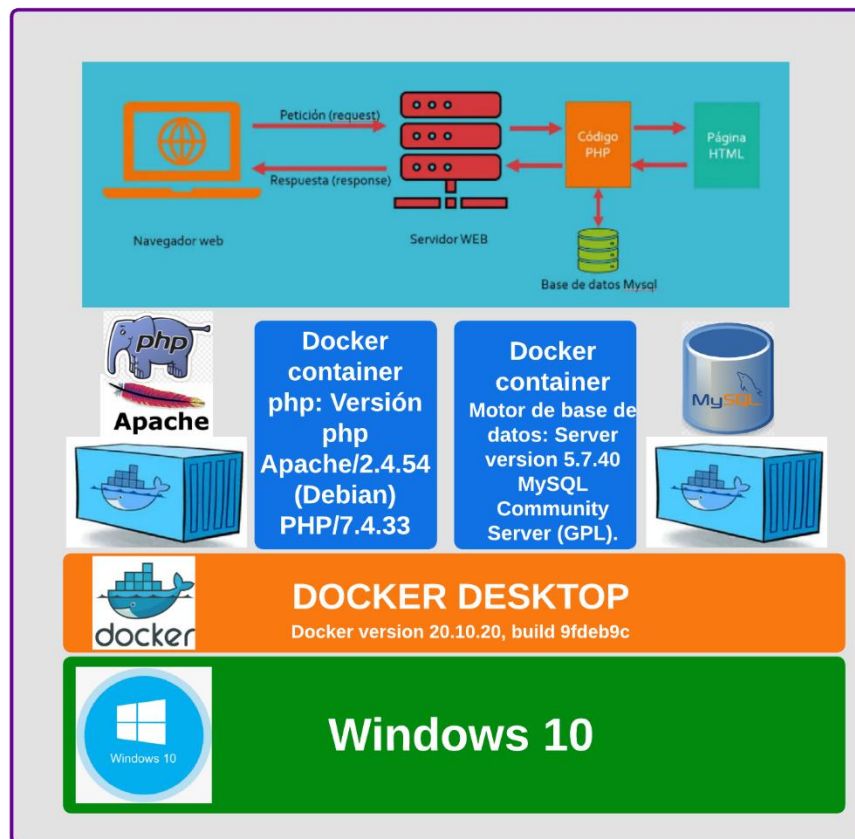


Ilustración 1 Diagrama de arquitectura general https://lucid.app/lucidspark/33d0cd90-6e3c-4549-b2d7-b642e9ec74e5/edit?viewport_loc=-8%2C-8%2C1835%2C917%2C0_0&invitationId=inv_29ce1902-0b5a-45e7-975f-b1c0c4a3834a

ROLES DEL EQUIPO

Rol	Descripción	Nombre
Administrador de proyecto	El administrador de proyecto les asignará a los analistas, la agenda con actividades a ser realizadas y sus fechas. Es claro que la asignación de actividades puede ir modificándose durante el proyecto.	Andres Villegas
Diseñador	Los diseñadores deben interactuar con los analistas para determinar la factibilidad del proyecto, y establecer los objetivos del sistema para un buen diseño. Los analistas deben permanecer en contacto estrecho con los diseñadores debido a que utilizarán la arquitectura del sistema. Los diseñadores deben poder ayudarle a los analistas.	Yeferson Noreña
Programador	Los analistas son apoyados por los programadores en el entendimiento y especificación de los requisitos de usuario y de software. Además, los apoyan en la construcción de prototipos rápidos	Alexander Rodriguez
Téster	Los analistas participan junto con los tésters en la revisión de los documentos de análisis de requisitos Debe revisar los documentos hechos por los analistas.	Sergio Figueroa
Documentador	Los analistas deberán entregarles la información que servirá para la documentación del sistema	Brayan Moreno

Tabla 1 Roles del proyecto

DESARROLLO DEL PROYETO

Descargamos la imagen de MySQL:5.7

```
PS C:\servicioweb> docker run -p 3306:3306 --name basedatos -v C:\servicioweb\database:/var/lib/mysql/ -e MYSQL_ROOT_PASSWORD=root -d mysql:5.7
Unable to find image 'mysql:5.7' locally
5.7: Pulling from library/mysql
9a0b9cd2dfe6: Pull complete
c637408ee7df: Pull complete
4c517093c276: Pull complete
301cc7d68c2a: Pull complete
17ca9bf9231a: Pull complete
9ae101e5c786: Pull complete
04baa409344e: Pull complete
f0b6015bf853: Pull complete
6005bb052ef8: Pull complete
99f303d57050: Pull complete
307a9a80c1df: Pull complete
Digest: sha256:0e3435e72c493aec752d8274379b1eac4d634f47a7781a7a92b8636fa1dc94c1
Status: Downloaded newer image for mysql:5.7
03fb52cd7ff548dbb9d45a6376bfd28a4e4d8ddb808cd90237cf8f5fd50e7c3
```

Ilustración 2 Imagen Docker pull mysql:5.7

Luego descargamos php:7.4-apache

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\servicioweb> docker pull php:7.4-apache
7.4-apache: Pulling from library/php
Digest: sha256:be775afc9eef8ae7babad2d4cf6176a6365768ac607e5d768e9a73bc92f7e214
Status: Image is up to date for php:7.4-apache
docker.io/library/php:7.4-apache
PS C:\servicioweb> █
```

Ilustración 3 Imagen php pull php:7.4-apache

Comprobamos imágenes instaladas:

```
PS C:\servicioweb> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mysql         5.7       eef0fab001e8   3 days ago    495MB
php           7.4-apache 51fd1ea74706   4 days ago    453MB
```

Ilustración 4 Verificación Imágenes consola

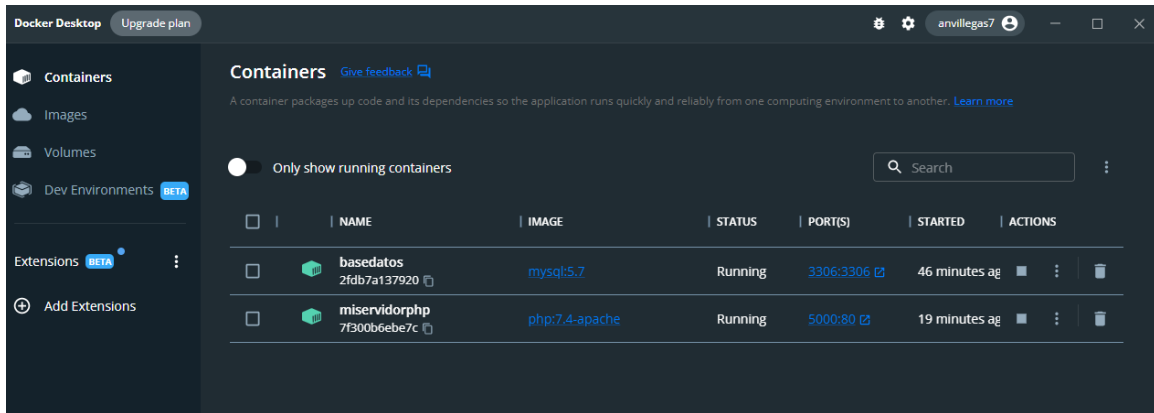


Ilustración 5 Vista desde el IDE de los contenedores:

Vamos a correr nuestro MySQL, con el comando abajo, necesitamos especificar los puertos con los que nos vamos a comunicar, el puerto por defecto para MySQL es 3306, el cual nos permite mapear la posición del contenedor.

```

vllle@LAPTOP-UU3QMVQL MINGW64 ~
$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
vllle@LAPTOP-UU3QMVQL MINGW64 ~
$ docker run -p 3306:3306 --name basedatos -v /c/temp/ServicioWeb/database:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=abc123 -d mysql:5.7
ac09d0c12b54d73fffe50ecabf1ef09ab26e016592317cc159fb48db68ac2734
vllle@LAPTOP-UU3QMVQL MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
ac09d0c12b54   mysql:5.7 "docker-entrypoint.s..." 11 seconds ago Up 9 seconds   0.0.0.0:3306->3306/tcp, 33060/tcp   basedatos

```

Ilustración 6 Puerto 3306 para conexión a mysql

Paso siguiente es crear nuestro contenedor de php con el puerto local 5000 y a nivel del contenedor el puerto 8080. Utilizamos la herramienta --link para que nuestro servidor web se conecte con la base de datos.

```

PS C:\servicioweb> docker run -p 5000:80 -v C:\servicioweb\src:/var/www/html --name miservidorphp -d --link basedatos php:7.4-apache
PS C:\servicioweb> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
7f300b6be7c    php:7.4-apache "docker-php-entrypoi..." 7 minutes ago Up 7 minutes   0.0.0.0:5000->80/tcp, 0.0.0.0:3306->3306/tcp   miservidorphp
2fdb7a137920    mysql:5.7 "docker-entrypoint.s..." 2 hours ago   Up 35 minutes   0.0.0.0:3306->3306/tcp, 33060/tcp   basedatos

```

Ilustración 7 Puerto 5000:80 para conexión a php

Creamos un volumen local para copiar del host al contenedor y viceversa.

Luego ejecutamos desde el browser localhost:5000 y vemos que nuestro servidor php Apache esta funcionando como se muestra abajo.

localhost:5000

PHP Version 7.4.33

System	Linux 7500b5be7c 5.10.102.1-microsoft-standard-WSL2 #1 SMP Wed Mar 2 00:30:59 UTC 2022 i86_64
Build Date	Nov 3 2022 17:25:59
Configure Command	'./configure' '--build=i86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-ehash' '--with-pcre' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=lib' '--with-sqlite3=lib' '--with-curl' '--with-com' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-pear' '--with-libdir=lib64' '--with-icu' --disable-cgi' '--with-asm=2' 'build_alias=i86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902 NTS
PHP Extension Build	API20190902 NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	ftp, ftps, compress.zlib, php, file, glob, data, http, https, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, convert.com.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine
Zend Engine v3.4.0, Copyright (c) Zend Technologies

zendengine

Configuration
apache2handler

Apache Version	Apache/2.4.54 (Debian)
Apache API Version	20120211
Server Administrator	webmaster@localhost

Ilustración 8 Conexión a php Apache localhost:5000

Ahora verificamos que la base de datos este corriendo:

```
bash-4.2# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
-> show databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'show databases' at line 2
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.03 sec)
```

Ilustración 9 Base de datos funcional

Ingresamos al servido php “miservidorphp”, para instalar la extensión de MySQL.


```

PS C:\servicioweb> docker exec -i -t miservidorphp /bin/bash
root@f91138b39f32:/var/www/html# ls
AccionCarta.php  Pagos.php          agregar_producto_final.php  css          historiaCliente.php  login.html      login_admin.html  producto_detalle.php  restarStock.php
Configuracion.php  Pagos2.php        buscador.php               editarProducto.php  img            login.php        login_admin.php    productos.php         tmp
La-carta.php      VerCarta.php      cargarImagen.php           eliminar_producto.php  index.php      loginOut.php     modulo_administrador.php  registro.php         web
OrdenExitos.php   agregar_producto.php  conn.php                   fonts         js              loginOutAdmin.php  producto.php        registro_solicitud.php
root@f91138b39f32:/var/www/html# docker-php-ext-install mysql
Configuring for:
PHP Api Version:      20190902
Zend Module Api No:   20190902
Zend Extension Api No: 320190902
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for a sed that does not truncate output... /bin/sed
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for cc... cc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether cc accepts -g... yes
checking for cc option to accept ISO C89... none needed

```

Ilustración 10 Configuración extensión MySQL dentro de Php

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| tienda_zapatos |
+-----+
5 rows in set (0.03 sec)

```

Ilustración 11 creamos la base de datos en el contenedor

Agregamos nuestro código php en el archivo Configuración php

```

<?php
//DB detalles
$dbHost = 'basedatos:3306';
$dbUsername = 'root';
$dbPassword = 'abc123';
$dbName = 'tienda_zapatos';

//Crear conexion a la base de datos
$db = new mysqli($dbHost, $dbUsername, $dbPassword, $dbName);

if ($db->connect_error) {
    die("No hay Conexion con la base de datos: " . $db->connect_error);
} else {
    echo "Connected to MySQL server successfully!";
}
?>

```

First commit to git hub <https://github.com/villegas984/IntegracionContinua.git>

```
$ git log --oneline
```

05fd3d8 (HEAD -> AndresV, origin/AndresV) 1st commit branch AndresV 11/08/22 inicio proyecto

Logramos una conexión exitosa.

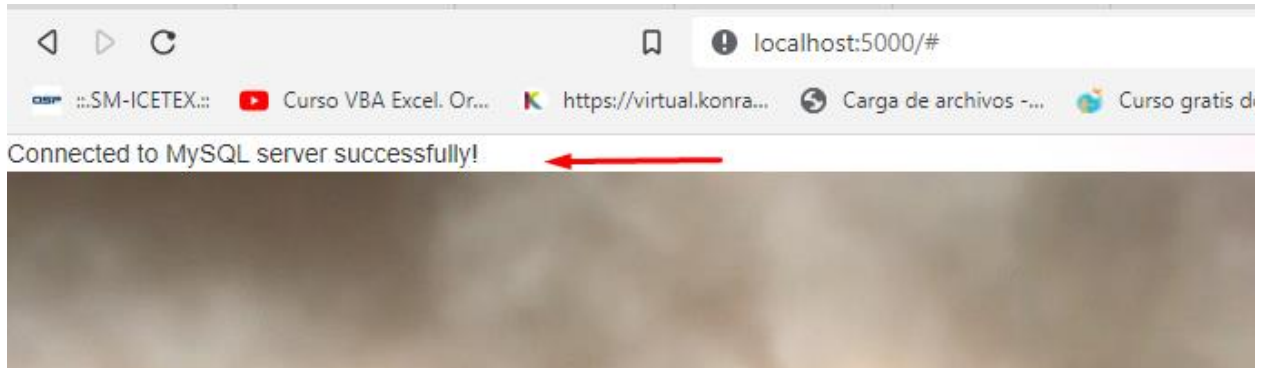


Ilustración 12 conexión a la página

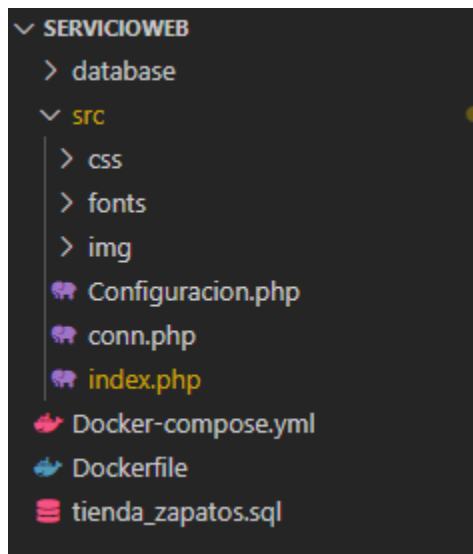


Ilustración 13 Estructura actual de nuestros código

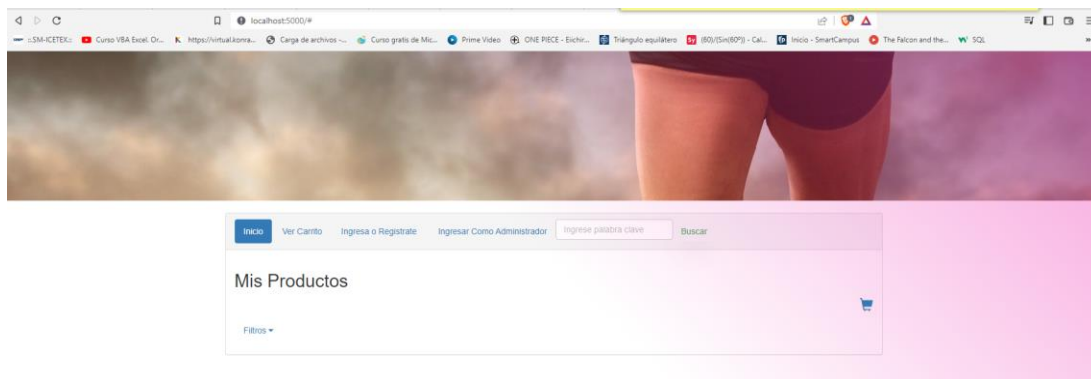


Ilustración 14 Vista inicial de la página

LINK DEL VIDEO

[Expo v1.mp4](#)

<https://drive.google.com/drive/folders/1DNF2CxVmxljVBEaOwyPLvO4rE1Cz29Nf>