Tabla de contenido

[Crear las tablas usuarios y productos en MySQL usando el modelo E/R proporcionado. 2](#_Toc173700335)

[1. Abrir XAMPP y habilitar apache y MySQL 2](#_Toc173700336)

[2. Abrir en un navegador http://localhost:8080/phpmyadmin/ 2](#_Toc173700337)

[3. Hacer clic en bases de datos 2](#_Toc173700338)

[4. Ponerle nombre y hacer clic en crear (api) 2](#_Toc173700339)

[5. Luego hacer clic en SQL en la parte superior 2](#_Toc173700340)

[6. Escribir el siguiente script SQL para crear las 2 tablas y la relación: 2](#_Toc173700341)

[7. Hacer clic en continuar y con eso ya tenemos la base creada con las 2 tablas. 2](#_Toc173700342)

[Desarrollar web services en PHP para realizar operaciones CRUD en ambas tablas. 2](#_Toc173700343)

[1. Configuración el servidor web 2](#_Toc173700344)

[2. Crea el directorio del proyecto 3](#_Toc173700345)

[3. Instalación de Dependencias 3](#_Toc173700346)

[4. Implementación de Web services 3](#_Toc173700347)

[5. Prueba con Postman 4](#_Toc173700348)

[Consumo de Web Services desde Android 5](#_Toc173700349)

[1. Configuración del proyecto 5](#_Toc173700350)

[2. Pruebas 6](#_Toc173700351)

[Ejecución del proyecto 6](#_Toc173700352)

[1. Se descarga y descomprime los archivos 6](#_Toc173700353)

[2. Se abre xampp y alza apache y MySQL 6](#_Toc173700354)

[3. Se crea la base y las tablas con el script proporcionado 6](#_Toc173700355)

[4. Luego configura el virtual host como indica arriba 6](#_Toc173700356)

[5. Cambio de ip en Android Studio 6](#_Toc173700357)

# Crear las tablas usuarios y productos en MySQL usando el modelo E/R proporcionado.

## Abrir XAMPP y habilitar apache y MySQL

## Abrir en un navegador <http://localhost:8080/phpmyadmin/>

**Nota: El 8080 es el puerto utilizado**

## Hacer clic en bases de datos

## Ponerle nombre y hacer clic en crear (api)

## Luego hacer clic en SQL en la parte superior

## Escribir el siguiente script SQL para crear las 2 tablas y la relación:

CREATE TABLE usuarios (

id INT(11) NOT NULL AUTO\_INCREMENT,

nombre VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL,

password VARCHAR(100) NOT NULL,

PRIMARY KEY (id)

) ENGINE = InnoDB;

CREATE TABLE productos (

id INT(11) NOT NULL AUTO\_INCREMENT,

nombre VARCHAR(100) NOT NULL,

descripcion TEXT NOT NULL,

precio DECIMAL(10,2) NOT NULL,

id\_usuario INT(11) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id)

) ENGINE = InnoDB;

## Hacer clic en continuar y con eso ya tenemos la base creada con las 2 tablas.

# Desarrollar web services en PHP para realizar operaciones CRUD en ambas tablas.

## Configuración el servidor web

1. Instalar XAMPP
2. Colocar el proyecto en el directorio “htdocs” de XAMPP

Ruta: C:\xampp\htdocs\

## Crea el directorio del proyecto

1. Crear una carpeta en htdocs llamada api
2. Ingresa desde el cmd a la carpeta

## Instalación de Dependencias

1. Ejecutar el siguiente comando para instalar Framework PHP como Slim

composer require slim/slim "^4.0"

composer require slim/psr7

1. En caso de no tener instalado composer
2. Instala Composer

<https://getcomposer.org/download/>

1. Verifica si se instaló , abre un terminal de cmd

composer –-version

## Implementación de Web services

1. Estructura del proyecto

Crea un directorio public y dentro un archivo index.php (contiene los endpoints)

1. Configura Apache

Crea el archivo .htaccess en la carpeta public con el siguiente contenido:

# Activar el módulo de reescritura

<IfModule mod\_rewrite.c>

RewriteEngine On

# Redirigir todas las solicitudes al archivo index.php

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteRule ^ index.php [L]

</IfModule>

1. Configura el Virtual Host

Ingresa a la carpeta de apache luego conf/extra/httpd-vhosts.conf (archivo para editar) y escribe el siguiente codigo

<VirtualHost \*:8080>

ServerName localhost

DocumentRoot "C:/xampp/htdocs/api/public"

<Directory "C:/xampp/htdocs/api/public">

AllowOverride All

Require all granted

</Directory>

</VirtualHost>

Editar el puerto y el nombre del ServerName si es necesario y que DocumentRoot apunte a la carpeta de public

1. Reinicia Apache

## Prueba con Postman

Si necesita probar los endpoints, lo puede hacer con Postman, ingresa

1. Crear una nueva solicitud

* Click new -> HTTP
* Seleccionar el método http adecuado para tu endpoint (post o get)
* Insertar la url del endpoint ejemplo: <http://localhost:8080//api/productos/read/1>
* Parámetros: Si el endpoint requiere parámetros, agrégales:

Cuerpo (Body): Para métodos como POST , puedes necesitar enviar datos en el cuerpo de la solicitud. Ve a la pestaña "Body" y elige el formato adecuado (raw->JSON). En el caso de los GET en la ruta agrega el parámetro como en el ejemplo anterior.

Encabezados (Headers): Si tu API requiere encabezados específicos (como tokens de autenticación), agrégales en la pestaña "Headers".

* En este caso se necesita agregar

Key: Content-Type

Value: application/json

* Haz clic en el botón "Send" para enviar la solicitud al servidor.

# Consumo de Web Services desde Android

## Configuración del proyecto

* 1. Crear un nuevo Proyecto
  2. Agregar Dependencias en el archivo build.gradle (Module :app)

implementation 'com.squareup.retrofit2:retrofit:2.9.0'

implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

* 1. Crear una carpeta llamada “models” para los modelos User, Product y ApiResponse
  2. Crear una carpeta llamada “network” y crea una interface para la Api que va a consumir los endpoints
  3. Configura Retrofit en la carpeta “network”

private static final String BASE\_URL = "http://192.168.20.108:8080/"; // Cambia esto a la URL de tu servidor

* 1. Crear una carpeta ui para las Activity de Product y User
  2. En la carpeta res crea un directorio “layout” y agrega los activity.xml que conlleva los diseños
  3. En la carpeta res en el directorio “xml” crea un archivo .xml llamado “network\_security\_config.xnl): Este archivo permite configurar la política de seguridad de red de la aplicación, especificando las reglas sobre el tráfico de red que la aplicación puede realizar.

Nota: Cambiar <domain includeSubdomains="true">192.168.20.108</domain> esta linea por la ip de su red.

* 1. En el archivo de AndroidManifest.xml se tiene que declara los componentes de la aplicación como las actividades, tambien define los permisos que la aplicación necesita.
* Dentro del elemento application se agrega la siguiente linea

android:networkSecurityConfig="@xml/network\_security\_config" : Esta linea especifica la configuración de seguridad de red utilizando el archivo

* Se agrega los activity para MainActivity(principal), UserActivity y ProductActivity
* Se agrega esta linea <uses-permission android:name="android.permission.INTERNET" /> -> declara que la aplicación necesita permiso para acceder a Internet

## Pruebas

* 1. Se conecta el dispositivo físico por medio USB o al emular de Android

# Ejecución del proyecto

## Se descarga y descomprime los archivos

## Se abre xampp y alza apache y MySQL

## Se crea la base y las tablas con el script proporcionado

## Luego configura el virtual host como indica arriba

## Cambio de ip en Android Studio

Para el aplicativo se hace el cambio de ip en los archivos

network\_security\_config.xml en el directorio res/xml

RetrofitClient en el directorio java/network