|  |
| --- |
| Interfaz de usuario gráfica  Descripción generada automáticamenteBenemerita Universidad Autonoma de Puebla  Facultad de Ciencias de la Computación  Servicios Web  Proyecto YHLQMDLG |
| **Equipo:**   * **Tomás García García 201936947** * **Gerardo Hernández Fernández 201920383**   **Profesor:**   * **M.C. Juan Carlos Conde R.** |

# Resumen

# Una API-REST (Application Programming Interface - Representational State Transfer) es una arquitectura de software para la construcción de servicios web. Se basa en el protocolo HTTP y se utiliza para crear aplicaciones web que se comunican entre sí a través de una interfaz de programación de aplicaciones (API). Los recursos se representan mediante URLs (Uniform Resource Locators) y se accede a ellos a través de los métodos HTTP estándar, como GET, POST, PUT y DELETE. Los datos se transmiten en formato JSON (JavaScript Object Notation) o XML (Extensible Markup Language). Las API-REST permiten que diferentes sistemas y aplicaciones se comuniquen entre sí de manera eficiente y estandarizada, lo que facilita la integración de sistemas y la creación de nuevas aplicaciones. Las API-REST también son escalables y flexibles, lo que las hace ideales para aplicaciones web modernas que requieren una alta disponibilidad y un alto rendimiento.

Por otro lado, Slim es un micro-framework para PHP que permite la creación de aplicaciones web y APIs RESTful de manera rápida y sencilla. Con Slim, es posible crear aplicaciones ligeras y escalables con una mínima cantidad de código y sin la necesidad de una compleja configuración. Está diseñado para ser lo más simple posible, lo que lo hace ideal para proyectos pequeños y medianos que no requieren de todas las características y funcionalidades de un framework completo.

Slim utiliza un enfoque modular para la creación de aplicaciones, lo que significa que se pueden agregar fácilmente funcionalidades adicionales mediante la instalación de complementos o middleware. Esto permite que los desarrolladores puedan adaptar Slim para satisfacer las necesidades específicas de sus proyectos. Además, Slim es altamente personalizable y se integra fácilmente con otras herramientas y bibliotecas de PHP. En general, Slim es una excelente opción para aquellos que buscan un marco de trabajo rápido, sencillo y altamente personalizable para el desarrollo de aplicaciones web y APIs RESTful en PHP.

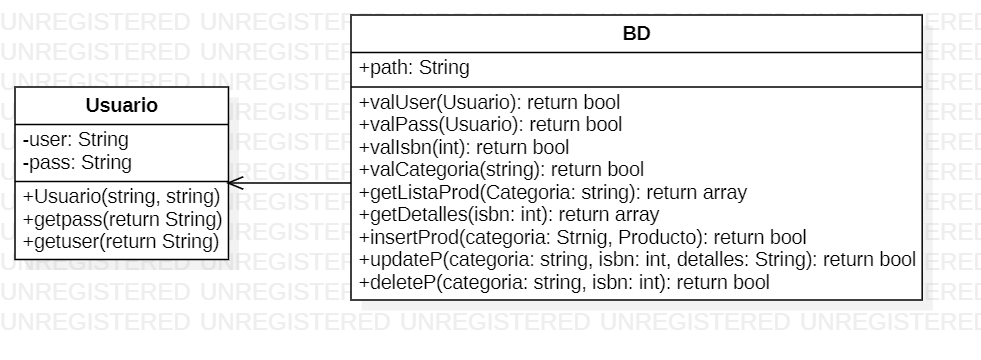
Slim es un microframework que se utiliza comúnmente para crear API RESTful en PHP, es compatible con API RESTful ya que proporciona las características necesarias para implementar las funciones principales de este tipo de API, por ejemplo:

* La capacidad de manejar solicitudes HTTP POST, GET, PUT y DELETE.
* La capacidad de enviar y recibir datos en diferentes formatos como JSON, XML, etc.

En resumen, Slim es una excelente opción para aquellos que buscan una forma sencilla de construir API RESTful en PHP, ya que proporciona todas las características necesarias para implementar este tipo de API de manera eficiente y esalable.

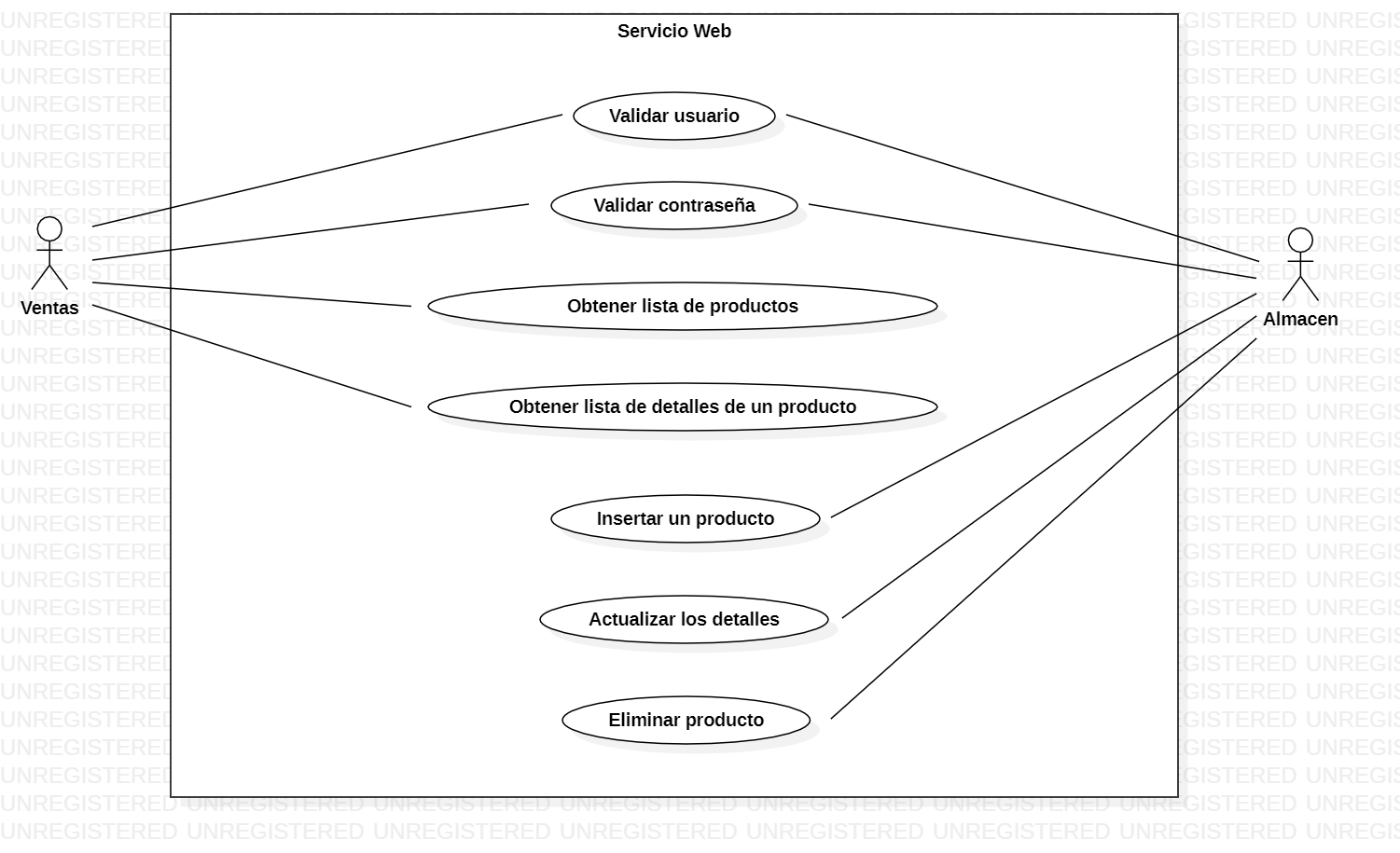
# Analisis y Diseño

# Diagrama de Clases



Como partida decidimos que era necesario la creación de 2 clases, la clase **Usuario,** la clase **BD.** La clase **Usuario** tiene como punto almacenar la información capturada del usuario, cada vez que se llame un método, se requerirá la “creación” de un nuevo usuario, con sus respectivos parámetros, estos se comparan con la base de datos y se valida la autorización. Por otro lado **BD**, se encarga de controlar todas las solicitudes entrantes, la validación de los usuarios y contraseñas, y el traspaso de los JSON entre el cliente y la base de Datos.

# Diagrama de Casos de Uso



De igual manera en el caso de uso, en este caso se ubicaron los roles posibles para los diferentes tipos de usuario que existen. Dependiendo de la acción que deba hacer el usuario, son los posibles eventos que pueden suceder.

Como vemos en el diagrama, una persona con el rol de ventas puede realizar ciertas tareas como hacer consultas. Al mismo tiempo, hay tareas que se efectúan para poder autorizar estas acciones, como es el caso de validar un usuario o validar una contraseña. En el Rol de almacén, pueden existir otro tipo de eventos, ya no solo consultas, si no que un almacenista puede hacer inserciones, modificaciones, o reducciones, en el inventario.

# Proceso de pruebas con POSTMAN

Prueba getListProd con Headers:

Captura de pantalla de computadora

Descripción generada automáticamente

Prueba getDetalles con Headers:

Captura de pantalla de computadora

Descripción generada automáticamente

Prueba inserProd con Headers:

Interfaz de usuario gráfica, Texto

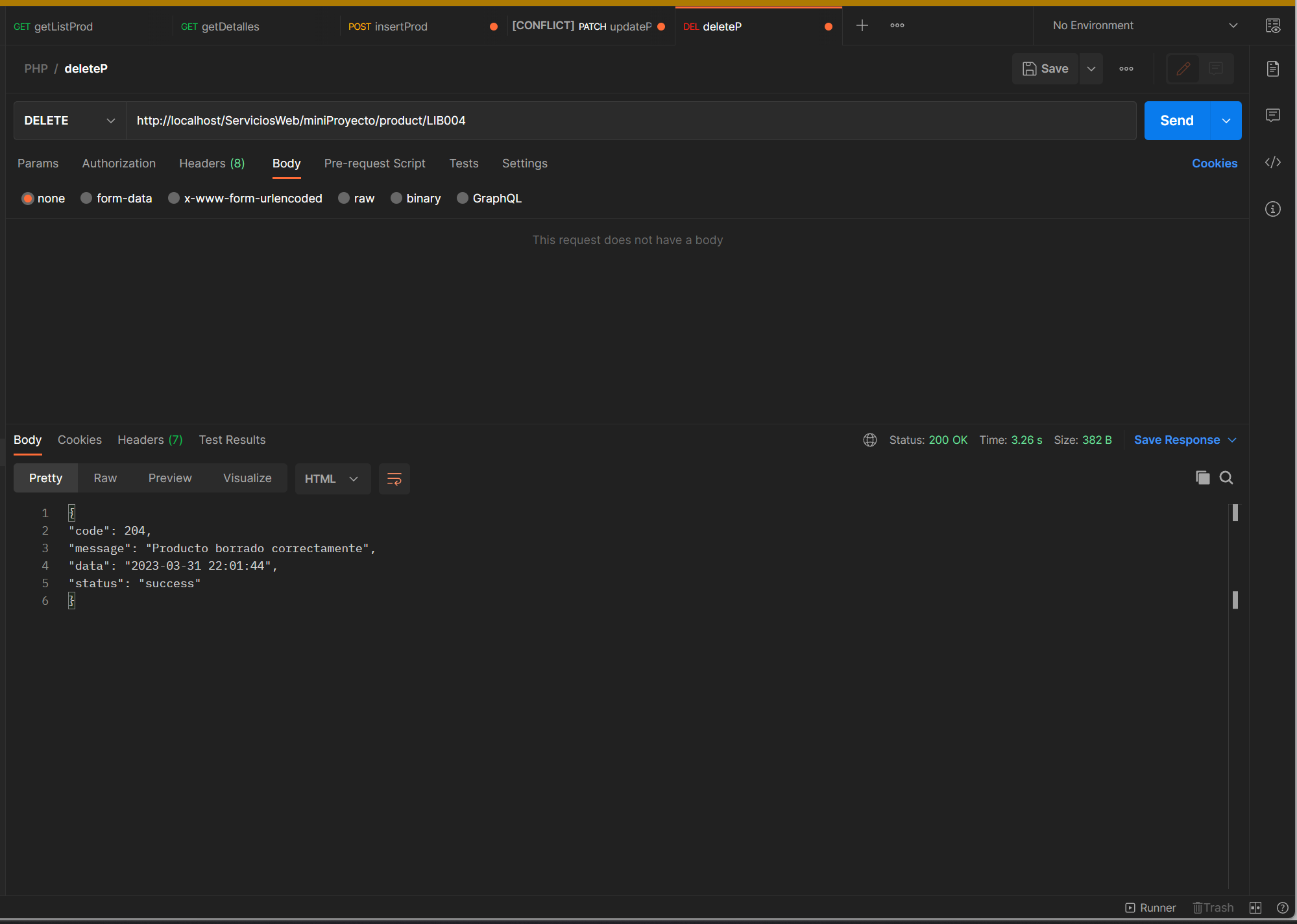
Descripción generada automáticamente

Prueba updateP con Headers:

Captura de pantalla de computadora

Descripción generada automáticamente

Prueba deleteP con Headers:



Proceso de pruebas con Cliente

Prueba getListProd con Headers:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Prueba getDetalles con Headers:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Prueba inserProd con Headers:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Prueba updateP con Headers:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Prueba deleteP con Headers:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente