

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



Plataforma sobre servicio de lenguaje de señas
Características Técnicas

Presentado por:
Chañi pauccar beckham vieri
Docente:
Ing. Fred Torres Cruz
Semestre:
7mo
Curso:
INGENIERIA DE SOFTWARE I

PUNO - PERÚ
2024

Índice

1. Arquitectura en Capas de la aplicación	3
2. Flujo de Servicios	3
3. Conclusión	4

1. Arquitectura en Capas de la aplicación

■ Capa de Presentación (Frontend)

- Responsabilidad: Gestionar la interacción con el usuario final, proporcionar la interfaz de usuario y manejar las solicitudes de los usuarios.
- Componentes:
 - Vistas: Archivos Blade de Laravel que renderizan la interfaz de usuario.
 - Controladores: Manejan las solicitudes HTTP entrantes, preparan los datos necesarios y devuelven las vistas correspondientes. Recursos Estáticos: Archivos CSS, JavaScript, imágenes, etc.

■ Capa de Aplicación (Lógica de Negocio)

- Responsabilidad: Contener la lógica de negocio central de la aplicación, coordinar las operaciones entre la capa de presentación y la capa de datos.
- Componentes:
 - Servicios: Clases que implementan la lógica de negocio específica. Por ejemplo, un servicio que maneja las compras de servicios de aprendizaje de lenguaje de señas
 - Facades: Proporcionan una interfaz sencilla para interactuar con las capas subyacentes, a menudo utilizadas para encapsular la lógica de negocio compleja.

■ Capa de Persistencia (Acceso a Datos)

- Responsabilidad: Manejar la interacción con la base de datos y otros sistemas de almacenamiento de datos.
- Componentes:
 - Modelos Eloquent: Representan las tablas de la base de datos y proporcionan métodos para interactuar con ellas.
 - Repositorios: Clases que encapsulan la lógica para acceder a los datos, proporcionando una capa de abstracción sobre los modelos Eloquent.

■ Capa de Infraestructura

- Responsabilidad: Proporcionar servicios de apoyo necesarios para la aplicación, como la gestión de sesiones, autenticación, colas de trabajo, y otros servicios de sistema.
- Componentes:
 - Middleware: Manejan las solicitudes HTTP antes de que lleguen a los controladores (autenticación, verificación de roles, protección CSRF, etc.).
 - Servicios de Integración: Clases que manejan la comunicación con servicios externos (pasarelas de pago, servicios de correo electrónico, APIs externas).

2. Flujo de Servicios

Registro y Autenticación de Usuarios

- Los usuarios (estudiantes y docentes) se registran proporcionando su información básica (nombre, correo electrónico, contraseña, etc.).

- Confirmación de registro a través de un enlace enviado por correo electrónico.

Gestión de Perfiles

- Los usuarios pueden actualizar su información personal, como nombre, foto de perfil y otros detalles relevantes.
- Los usuarios pueden ver su perfil y revisar información relevante, como sus compras y clases registradas.

Exploración y Compra de Servicios

- Los estudiantes pueden navegar a través de una lista de servicios de aprendizaje disponibles.
- Filtros y búsquedas para encontrar servicios específicos (por nivel, tipo de clase, etc.).

Gestión de Clases por Parte de los Docentes

- Los docentes pueden crear y programar clases, especificando detalles como el título, la descripción, la fecha y hora, y el precio.
- Posibilidad de programar clases individuales o en grupos.
- Los docentes pueden ver la lista de estudiantes inscritos en sus clases y gestionar la asistencia.

Participación en Clases por Parte de los Estudiantes

- Los estudiantes pueden registrarse en clases específicas después de la compra del servicio.
- Los estudiantes pueden registrarse en clases específicas después de la compra del servicio.

3. Conclusión

Este trabajo presenta los requerimientos de la aplicación web destinada a la venta de servicios de aprendizaje de lenguaje de señas. La aplicación ha sido desarrollada utilizando Laravel y MySQL, aprovechando las capacidades robustas y escalables de estas tecnologías.

La definición clara de los requisitos funcionales y no funcionales ha sido esencial para guiar el desarrollo y asegurar que la plataforma cumpla con las expectativas y necesidades de los usuarios. Los requisitos funcionales abarcan desde la gestión de usuarios y servicios hasta la programación y asistencia a clases, garantizando que todas las funcionalidades críticas estén bien cubiertas. Por su parte, los requisitos no funcionales aseguran que la aplicación sea segura, rápida, accesible y capaz de manejar un crecimiento en la base de usuarios y transacciones.