

PROYECTO MOVILIDAD TAXISEAN : FASE DETALLADA

PROGRAMACION ORIENTADA A OBJETOS

González Oscar, Ovalle Valery, Ñustas
Cristian

{ogonzal25567, vovalle82473,
cnustes78568}@universidadean.edu.co

*Universidad EAN
Bogotá, Colombia*

RESUMEN

El presente proyecto informe tiene como objetivo detallar el proyecto de desarrollo de un software basado en programación Orientada a Objetos (POO) para así facilitar el servicio de pedido de taxis. El objetivo de este software es crear una plataforma eficiente y fácil de usar para los usuarios que permita solicitar taxis de manera rápida y segura, planear la ruta y además esta pagina web contara con un sistema de pago integrado. El software propuesto contara con una interfaz intuitiva y amigable para los usuarios, estos podrán registrarse en la plataforma, ingresar su ubicación actual y el destino al que desea viajar, asi mismo podrá seleccionar el tipo de carro requerido. Además, se implementarán funciones como el tiempo estimado de llegada, la ubicación del taxi y diferentes métodos de pago. Para lograr este proyecto, se utilizará Python como lenguaje de programación, entre otras tecnologías. También se considerará almacenar la información de los usuarios y los registros de los viajes en una base de datos.

Palabras Clave *Eficiente – Interfaz Intuitiva – Base de Datos – Usuario – Implementación*

ABSTRACT

The objective of this report project is to detail the development project of a software based on Object Oriented Programming (OOP) to facilitate the taxi ordering service. The objective of this software is to create an efficient and easy-to-use platform for users that allows them to request taxis quickly and safely, plan the route and this website will have an integrated payment system. The proposed software will have an intuitive and user-friendly interface, they will be able to register on the platform, enter their current location and the destination they want to travel to, and they will also be able to select the type of car required. In addition, functions such as estimated time of arrival, taxi location and different payment methods will be implemented.

To achieve this project, Python will be used as a programming language, among other technologies. Storing user information and travel records in a database will also be considered.

Key Words *Efficient – Intuitive Interface – Database – User – Implementation*

1. INTRODUCCIÓN

El presente proyecto tiene como objetivo desarrollar un software en POO para facilitar el servicio para solicitar un taxi. Mediante el uso de la POO y las últimas tecnologías, este proyecto busca crear una plataforma intuitiva y fácil de usar que permita a los usuarios solicitar taxis de manera rápida y confiable.

Se identificarán las entidades principales del sistema, como usuarios, taxis y viajes, y se crearán clases y objetos correspondientes para cada entidad. Además, se implementarán patrones de diseño de software adecuados para garantizar un diseño eficiente y flexible.

El desarrollo del proyecto se llevará a cabo en diferentes etapas, incluyendo análisis de requisitos, diseño de la arquitectura, implementación, pruebas y despliegue. El sistema se ha desarrollado utilizando los principios de POO en Python, lo que permite una mejor organización del código y una mayor facilidad para el mantenimiento y la escalabilidad del sistema.

En este proyecto, se muestra cómo la programación orientada a objetos puede utilizarse para resolver problemas del mundo real, mejorando la eficiencia y la experiencia del usuario. Esperamos que este proyecto sea de utilidad para la comunidad de usuarios de Taxis y sirva como ejemplo de cómo la tecnología puede mejorar los servicios públicos.

2. DESARROLLO

- **IDENTIFICACIÓN DEL PROBLEMA**

En el contexto del servicio público de movilidad como lo es el taxi, encontramos que hay una necesidad enorme de una mejor; por lo cual nuestro proyecto surge como una respuesta a esta necesidad. Aunque como ya sabemos el servicio de taxi ha estado disponible por muchos años este aun asi presenta problemáticas que dificultan su eficiencia y comodidad. Entre los principales factores que intentamos solucionar con nuestro software son:

- Proceso de solicitud: Este proceso suele ser muy tedioso y difícil así ya que los usuarios tienden a enfrentar dificultades para encontrar taxis disponibles en el lugar y momento necesario. Por lo tanto, nuestro software permitirá al usuario pedir un taxi en cualquier lugar y este tendrá un corto periodo de confirmación.
- Seguridad y confiabilidad: La seguridad es muy importante a la hora de utilizar taxis, los usuarios pueden tener dudas acerca del perfil del conductor y lo fiable que este podría ser. Por eso para nuestro software es de vital importancia contar con un sistema confiable que brinde tranquilidad a los usuarios.

- **OBJETIVOS**

- ✓ Desarrollar un sistema para la solicitud de taxis que permita a los usuarios solicitar un taxi en el momento o lugar necesario, para que los usuarios estén más seguros y además tengan un periodo corto de espera de confirmación
- ✓ Ofrecer a los usuarios una plataforma intuitiva y eficiente para la gestión de solicitud de taxi, además que este cuente con una óptima gestión de usuario y perfiles
- ✓ Garantizar la seguridad y privacidad de los usuarios mediante la implementación de medidas de protección de datos, como la encriptación de información personal y la autenticación de usuarios.
- ✓ Mejorar la calidad del servicio que se ofrece a los ciudadanos y la percepción de la ciudadanía sobre el transporte público en general, mediante la implementación de un sistema eficiente, confiable y fácil de usar.

- **REQUERIMIENTOS**

Requerimientos Funcionales:

- El sistema debe permitir el registro de usuarios en la página web.
- El sistema debe permitir que el usuario proporcione información personal para poder realizar los detalles de pago
- El sistema debe permitir una planificación de rutas para los desplazamientos de los usuarios
- Sistema de seguimiento de taxis para garantizar la seguridad y el cumplimiento de las reglas de uso
- Integración de un sistema de pago seguro y confiable
- El sistema permite que el vacante a cualquier cargo pueda subir la documentación necesaria
- El sistema permite que el administrador agregue o elimine un empleado.
- El sistema permite actualizar la información de los empleados
- El sistema permite generar notificaciones y/o alertas para indicar que el taxi ha llegado.
- El sistema permite visualizar efectivamente el tiempo aproximado del taxi para llegar a su destino.
- El sistema de movilidad debe tener un sistema de atención al cliente para resolver problemas y preguntas de los usuarios

Requerimientos no funcionales:

- El sistema permite manejar una alta cantidad de usuario sin saturarse, además permite procesar las transacciones rápidamente

- El sistema está disponible para los usuarios en cualquier momento y cuenta con una cantidad de tiempo de inactividad
- El sistema está protegido contra amenazas externas, como el robo de datos.
- El sistema debe tener una interfaz amigable con el usuario, que sea muy intuitiva
- El sistema es adaptable al dispositivo y no va a afectar el rendimiento del computador
- El sistema permite la actualización.

• **ARQUITECTURA DEL SISTEMA**

Para la arquitectura del sistema del proyecto de TASXISEAN planteamos el siguiente diagrama de clases, para poder lograr esto analizamos la situación y los requerimientos que este necesita.

Finalmente, estructuramos el diagrama de clases de la siguiente manera:

Primero que todo estructuramos la super clase Person (Persona) la cual tiene como atributos name(nombre), age (edad), gender (género) y DNI (número de identificación), las personas podrán acceder y salir de la cuenta (login and log out). De esta super clase heredan 3 clases, las cuales son Passenger (pasajero), Driver(conductor) y admin (Administrador).

La clase pasajero hereda los atributos de la super clase Person (persona) y los atributos que posee son generateCard (generar una tarjeta), delete Card (eliminar tarjeta), recharge card (recargar tarjeta), request taxi (pedir taxi), cancel taxi (cancelar taxi) y pay Travel (pagar el viaje) con el metodo de pago que desee.

La clase conductor hereda de la clase persona, adicionalmente tiene los atributos idConductor, driver | license (licencia de conducir) y status de conductor; como métodos del conductor puede aceptar un viaje, cancelar un viaje, empezar y terminar viaje.

La clase Admin hereda de la clase persona y tiene un atributo id Admin, este administrador tiene los métodos eliminar conductor y añadir conductor, actualizar conductor, eliminar taxi , añadir taxi, actualizar taxi, eliminar pasajero, actualizar pasajero.

Como podemos observar la imagen 1 de diagrama de clases la clase pasajero y la clase conductor tienen una relación de asociación con la multiplicidad 1..* a 1, ya que el conductor puede tener uno o mas pasajeros y la clase conductor tiene una relación de asociación con el administrador con multiplicidad 1..* a 1 ya que muchos conductores pueden tener un administrador.

Otras relaciones que observamos es que el taxi depende del conductor, el taxi depende del pasajero.

La clase Taxi Card(tarjeta de Taxi) tiene como atributos información del pasajero, monto en la tarjeta y generar tarjetas, sus métodos son generar la tarjeta del taxi, eliminar la tarjeta, recargar tarjeta, pagar el viaje

Como último tenemos la clase taxi, esta clase tiene como atributos placa, estatus, capacidad, modelo y conductor y sus métodos asignar viaje, asignar conductor, empezar viaje y finalizar viaje.

• **DISEÑO INTERFAZ**

Taxi Card System

Generate Card	Recharge Card	Pay Travel	Delete Card
---------------	---------------	------------	-------------

Name:

Age:

Gender:

DNI:

Passenger ID:

Generate Card

Taxi Card System

Generate Card	Recharge Card	Pay Travel	Delete Card
---------------	---------------	------------	-------------

Passenger ID:

Amount to Recharge:

Recharge Card

Taxi Card System

Generate Card	Recharge Card	Pay Travel	Delete Card
---------------	---------------	------------	-------------

Passenger ID:

Travel Value:

Pay Travel

Taxi Card System

Generate Card

Recharge Card

Pay Travel

Delete Card

Passenger ID:

Delete Card