

| **1. Informe final Proyecto APT** |
| --- |
| El proyecto “TaxyPro” tiene como objetivo principal mejorar el servicio de radiotaxis en zonas donde las aplicaciones tradicionales no funcionan de manera eficiente, como las zonas periféricas y las llamadas "zonas rojas". Para ello, se ha diseñado e implementado una plataforma tecnológica que permite a los usuarios solicitar un taxi mediante una aplicación móvil, y a los conductores gestionar de manera eficiente sus solicitudes y rutas.  Durante el proceso de desarrollo, se tomaron decisiones clave en cuanto a la arquitectura de la aplicación, el tipo de tecnologías a utilizar (Flutter para el frontend y NestJS para el backend), y la implementación de funcionalidades críticas como la geolocalización en tiempo real, los pagos electrónicos y la seguridad en la creación de cuentas de usuario.  El desarrollo del proyecto se dividió en fases que incluyeron análisis de requerimientos, diseño, implementación, pruebas y ajustes finales. Cada una de estas fases fue monitoreada y ajustada según las necesidades que surgieron en el camino. |

| Nombre del proyecto | TaxyPro |
| --- | --- |
| Área (s) de desempeño(s) | **Desarrollo de Software y Aplicaciones:** El proyecto TaxyPro se centra en el desarrollo de una aplicación web y móvil (Android) para la gestión de servicios de taxi. Esta área de desempeño es fundamental en el perfil de egreso, ya que los ingenieros en informática y telecomunicaciones deben ser capaces de diseñar, desarrollar e implementar soluciones tecnológicas innovadoras.  **Gestión de Proyectos Tecnológicos:** La planificación, ejecución y control del proyecto TaxyPro implican competencias en gestión de proyectos, incluyendo la coordinación de equipos, la gestión de recursos y la supervisión del cumplimiento de plazos y objetivos. Esta área es crucial para asegurar la entrega exitosa de proyectos tecnológicos.  **Integración de Sistemas y Redes:** El proyecto también aborda la integración de diferentes sistemas y tecnologías, como APIs de geolocalización y sistemas de pago electrónico, asegurando la interoperabilidad y la seguridad de la información. Esta área de desempeño es esencial para los ingenieros que deben trabajar con múltiples plataformas y tecnologías. |
| Competencias | **Competencia Técnica en Desarrollo de Software:**   * **Diseño y Programación:** Capacidad para diseñar y programar aplicaciones utilizando lenguajes y frameworks modernos como JavaScript, HTML y CSS para la web, y Java, Flutter y kotlin para Android. Esto incluye la capacidad de crear interfaces de usuario intuitivas y eficientes, así como la implementación de funcionalidades de backend seguras y escalables. * **Pruebas y Mantenimiento**: Habilidad para realizar pruebas unitarias, de integración y de aceptación, asegurando que el software cumpla con los estándares de calidad. Además, experiencia en la gestión del ciclo de vida del software, incluyendo actualizaciones y correcciones continuas para mantener la estabilidad y rendimiento de la aplicación.   **Competencia en Gestión de Proyectos:**   * **Planificación y Organización:** Capacidad para planificar y organizar proyectos tecnológicos complejos, como el desarrollo de la plataforma “TaxyPro”, definiendo claramente los objetivos, el alcance del proyecto, los cronogramas, y la asignación eficiente de los recursos. * **Liderazgo y Trabajo en Equipo:** Habilidad para liderar equipos de desarrollo multidisciplinarios, fomentando la colaboración y la comunicación efectiva. Experiencia en la asignación de tareas, manejo de conflictos y seguimiento del progreso del equipo para garantizar la entrega de resultados de alta calidad.   **Competencia en Integración de Sistemas:**   * **Interoperabilidad:** Capacidad para integrar múltiples sistemas y tecnologías en la plataforma TaxyPro, asegurando la compatibilidad y eficiencia operativa entre la aplicación móvil, el backend, y los sistemas de terceros como los servicios de geolocalización y procesamiento de pagos. * **Seguridad de la Información:** Habilidad para implementar medidas de seguridad robustas, incluyendo el cifrado de datos y la autenticación segura, para proteger la integridad y confidencialidad de los datos de los usuarios. Experiencia en el cumplimiento de normativas y mejores prácticas en ciberseguridad. |

| **Contenidos del informe final** | |
| --- | --- |
| 1. Relevancia del proyecto APT | El proyecto TaxyPro se conecta directamente con el perfil de egreso del año 2020 de la carrera de Ingeniería en Informática y Telecomunicaciones del Duoc UC Puente Alto. Este perfil de egreso destaca la capacidad de los egresados para desarrollar soluciones tecnológicas innovadoras que respondan a necesidades reales del mercado y la sociedad. TaxyPro, al ser una plataforma que optimiza la gestión de servicios de taxi, aborda una necesidad crítica en el ámbito del transporte urbano, mejorando la eficiencia y la seguridad tanto para conductores como para pasajeros. |
| 2. Objetivos | **Objetivo General:** Desarrollar una plataforma tecnológica que optimice la gestión de servicios de taxi en la comuna de Puente Alto, mejorando la eficiencia operativa y la experiencia del usuario.  **Objetivos Específicos:**   1. Implementar un sistema de geolocalización en tiempo real para la asignación de taxis, mejorando la precisión y rapidez en la conexión entre pasajeros y conductores. 2. Desarrollar una interfaz intuitiva y amigable para usuarios y conductores, facilitando el uso de la plataforma y optimizando la interacción con las funcionalidades clave. 3. Integrar un sistema de pagos electrónicos seguro, garantizando la protección de los datos y transacciones de los usuarios. |
| 3. Metodología | Para el desarrollo del proyecto TaxyPro, se utilizó la metodología tradicional, que se divide en las siguientes fases:   1. **Planificación:** Se definieron los objetivos del proyecto, se identificaron los requisitos del sistema y se elaboró un plan de trabajo detallado. En esta fase, se realizó una estimación de tiempos y recursos necesarios para cada una de las actividades del proyecto, lo que permitió establecer un cronograma y las tareas correspondientes a cada miembro del equipo. 2. **Análisis:** Se llevó a cabo un análisis exhaustivo de las necesidades del mercado y de los usuarios, utilizando técnicas de investigación como entrevistas y encuestas. Además, se realizó un estudio de viabilidad técnica y económica, evaluando las tecnologías disponibles para implementar el sistema de manera eficiente. Se utilizaron herramientas como análisis FODA y estudio de costos para determinar la mejor estrategia tecnológica. 3. **Diseño:** Se diseñaron las arquitecturas del sistema utilizando el patrón Model-View-Controller (MVC) para la web y una arquitectura basada en Microservicios para el backend. Además, se diseñaron las interfaces de usuario con énfasis en la usabilidad y la escalabilidad del sistema. Se emplearon herramientas como Figma para los prototipos de diseño y se aseguraron los estándares de accesibilidad para usuarios con diferentes niveles de habilidad técnica. 4. **Implementación:** Se desarrollaron los módulos del sistema utilizando tecnologías modernas como JavaScript y Node.js para la web, junto con Java y Kotlin para el desarrollo de la aplicación móvil en Android. Se integran funcionalidades clave como la geolocalización en tiempo real y el procesamiento de pagos electrónicos seguros, utilizando APIs como Google Maps y Stripe. Además, se desarrollaron servicios RESTful para la comunicación entre el frontend y el backend, utilizando NestJS para la parte del servidor y MySQL como base de datos. 5. **Pruebas:** Se llevaron a cabo pruebas unitarias con herramientas como JUnit y Mocha para asegurar que cada componente funciona correctamente de manera independiente. También se realizaron pruebas de integración para validar la interacción entre los diferentes módulos del sistema. Finalmente, se implementaron pruebas de aceptación con usuarios finales para verificar la usabilidad y el cumplimiento de los requisitos del proyecto. 6. **Mantenimiento:** Se estableció un plan de mantenimiento para asegurar la continuidad operativa del sistema y su actualización periódica de acuerdo con las necesidades emergentes. Este plan incluye la monitorización del sistema, solución de problemas y la implementación de mejoras continuas en el rendimiento y la seguridad de la plataforma. |
| 4. Desarrollo | **Etapas o actividades desarrolladas:**   1. **Análisis de requisitos:** Se identificaron y documentaron los requisitos funcionales y no funcionales del sistema. Entre los requisitos funcionales se incluyeron la necesidad de un sistema de geolocalización en tiempo real, la integración de pagos electrónicos seguros, y la gestión de perfiles de usuarios. Los requisitos no funcionales incluyen la seguridad del sistema, la escalabilidad para futuros aumentos de usuarios y la velocidad de respuesta de la aplicación. 2. **Diseño del Sistema:** Se elaboraron los diagramas de arquitectura del sistema, diagramas de flujo de datos, y modelos de base de datos. Se diseñaron las interfaces de usuario utilizando herramientas de diseño UX/UI como Figma, asegurando una experiencia de usuario intuitiva tanto en la aplicación web como en Android. Los modelos de base de datos se desarrollaron con diagramas entidad-relación, optimizando la estructura de datos para facilitar las consultas y operaciones. 3. **Desarrollo:** Se implementaron los módulos del sistema utilizando tecnologías como HTML, CSS, JavaScript para la aplicación web, y Java/Kotlin para la aplicación Android. Se integraron APIs como Google Maps para la geolocalización y Stripe para los pagos electrónicos, asegurando una integración fluida y segura en ambos casos. 4. **Pruebas:** Se llevaron a cabo pruebas unitarias, de integración y de aceptación para asegurar la calidad del software. Se utilizaron herramientas como JUnit para las pruebas en la aplicación Android y Selenium para las pruebas de la aplicación web. Además, se realizaron pruebas de rendimiento para garantizar que el sistema pudiera soportar la carga prevista de usuarios sin afectar su rendimiento. 5. **Despliegue:** Se configuraron los servidores y se desplegó la aplicación en un entorno de producción utilizando servicios en la nube como AWS. El despliegue incluyó un pipeline de integración continua (CI/CD) para facilitar actualizaciones y parches, asegurando que el equipo pueda realizar mejoras sin interrumpir el servicio. 6. **Mantenimiento:** Se estableció un plan de mantenimiento que incluye la actualización periódica del sistema y la corrección de errores. Además, se estableció un sistema de monitoreo para anticipar problemas y planificar el escalado de recursos en función del crecimiento de usuarios.   **Dificultades y facilitadores:**   * **Facilitadores:** La utilización de frameworks y librerías modernas como React para la web y Retrofit para Android facilitó el desarrollo ágil de la aplicación. La colaboración efectiva del equipo mediante herramientas como Slack y Jira contribuyó al éxito del proyecto. Además, la adopción de metodologías ágiles permitió iteraciones rápidas y eficientes. * **Dificultades:** Se enfrentaron desafíos en la integración de APIs de terceros, especialmente en la sincronización de los datos de geolocalización en tiempo real con la asignación de conductores. Estas dificultades se abordaron mediante sesiones intensivas de debugging y optimización de código. Adicionalmente, se encontraron problemas en la optimización del rendimiento de la aplicación en dispositivos con recursos limitados, que se solucionaron mediante la implementación de estrategias de caching y reducción de la carga en la interfaz.   **Ajustes al software:**   * **Optimización de rendimiento:** Se realizaron ajustes en el código para mejorar la eficiencia de las consultas a la base de datos y la carga de la interfaz de usuario. * **Mejoras en la seguridad:** Se implementaron medidas adicionales de seguridad, como la encriptación de datos sensibles y la autenticación de dos factores. * **Corrección de errores:** Se solucionaron errores identificados durante las pruebas, mejorando la estabilidad y usabilidad del sistema.   **Abordaje de dificultades:**   * **Integración de APIs:** Se realizaron pruebas exhaustivas y se ajustaron las configuraciones para asegurar una integración fluida con servicios externos. * **Optimización de rendimiento:** Se llevaron a cabo revisiones de código y se implementaron técnicas de caching para mejorar la velocidad de respuesta de la aplicación. |
| 5. Evidencias | **Evidencias presentadas:**   * **Documentación técnica:** Incluye diagramas de arquitectura, modelos de datos y especificaciones de diseño. * **Código fuente:** Repositorios de GitHub con el código del proyecto, incluyendo commits detallados y comentarios. * **Resultados de pruebas:** Reportes de pruebas unitarias, de integración y de aceptación que demuestran el cumplimiento de los requisitos. * **Demostraciones funcionales:** Videos y capturas de pantalla que muestran el funcionamiento de la aplicación en diferentes escenarios.   **Evidencias de mayor valor:**   * **Resultados de pruebas:** Proveen una validación objetiva del correcto funcionamiento del sistema. * **Demostraciones funcionales:** Permiten visualizar de manera clara y directa cómo opera la aplicación, facilitando la comprensión de los resultados obtenidos. |
| 6. Intereses y proyecciones profesionales | **Reflexión sobre el aporte del Proyecto APT:** El desarrollo de TaxyPro ha permitido profundizar en conocimientos técnicos y habilidades prácticas en el ámbito de la ingeniería informática, especialmente en el desarrollo de aplicaciones móviles y web. Además, ha fomentado competencias en gestión de proyectos y trabajo en equipo.  **Proyecciones laborales:**   * **Intereses profesionales:** Continuar explorando el desarrollo de aplicaciones móviles y web, con un enfoque en la integración de tecnologías emergentes como la inteligencia artificial y el machine learning. * **Proyección laboral:** Después de completar el proyecto TaxyPro, se proyecta una carrera en el desarrollo de software, con especial interés en roles de liderazgo técnico y gestión de proyectos. |