**Autoevaluación 1.3 –**

**Proyecto Gasway**

Alumno: Ignacio Cisternas

Profesor: Francisco Juillet

Asignatura: Capstone

25/08/2024

Índice

[**Descripción de proyecto APT** 2](#_Toc175581011)

[**Relación del proyecto APT con las competencias del perfil de egreso** 3](#_Toc175581012)

[**Relación del proyecto-intereses profesionales** 5](#_Toc175581013)

[**¿Es factible a realizarse dentro de la asignatura?** 5](#_Toc175581014)

[**Propuesta metodológica** 5](#_Toc175581015)

[Plan de trabajo para el proyecto APT 6](#_Toc175581016)

[Propuesta de evidencias 6](#_Toc175581017)

# 

# **Descripción de proyecto APT**

El proyecto APT a presentar es una aplicación móvil desarrollada en React Native enfocada en la distribución de gas por parte de repartidores independientes en Chile. Este proyecto responde a la realidad local donde los distribuidores de gas no son directamente las grandes empresas como Gasco o Lipigas, sino personas independientes que adquieren el gas de estas grandes compañías y lo distribuyen a los consumidores finales.

El objetivo del proyecto APT es conectar a estos distribuidores independientes con los usuarios que necesitan servicios de distribución de gas de manera eficiente y conveniente. La aplicación busca digitalizar y optimizar el proceso de solicitud, asignación y entrega de gas, ofreciendo una plataforma donde tanto usuarios como distribuidores puedan interactuar de manera directa, rápida y confiable.

# **Relación del proyecto APT con las competencias del perfil de egreso**

El proyecto APT se alinea estrechamente con las competencias del perfil de egreso del programa académico, demostrando habilidades clave en pruebas de certificación, gestión de proyectos, modelado de datos, y desarrollo de software. A continuación, se detalla cómo este proyecto refleja estas competencias:

**1. Realizar Pruebas de Certificación de Productos y Procesos**

**Competencia:** Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria.

**Indicadores de calidad:**

* **1.1 Diseñar Pruebas de validación:** En el proyecto APT, se diseñan pruebas de validación tanto para los productos (aplicación móvil) como para los procesos (gestión de pedidos, procesamiento de pagos, etc.), utilizando metodologías reconocidas como TDD (Test-Driven Development).
* **1.2 Aplicar Pruebas de validación:** Las pruebas se aplican en cada fase del desarrollo, desde la integración de la API de Google Maps hasta la implementación de las pasarelas de pago, asegurando que todos los componentes funcionen correctamente antes del despliegue.
* **1.3 Desarrollar mejoras:** Los resultados de las pruebas se utilizan para iterar y mejorar continuamente el producto.

**2. Construir modelos de datos**

**Competencia:** Construir modelos de datos para soportar los requerimientos de la organización de acuerdo con un diseño definido y escalable en el tiempo.

**Indicadores de Calidad:**

* **3.1 Diseñar Modelos de datos:** Nuestro proyecto APT incorpora un diseño de base de datos relacional y no relacional (dependiendo de las necesidades), estructurado para manejar de manera eficiente la información de usuarios, distribuidores, pedidos, y transacciones financieras.
* **3.2 Implementar Modelos de datos:** La implementación del modelo de datos incluye la creación de esquemas de bases de datos que soportan las operaciones en tiempo real.

**3. Desarrollar una Solución de Software**

**Competencia:** Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos.

**Indicadores de Calidad:**

* **4.1 Construir una Solución de software:** El proyecto APT sigue una arquitectura modular que permite un desarrollo organizado y mantenible. Se emplean técnicas de desarrollo ágil como Scrum para facilitar la adaptación y mejora continua del software.
* **4.2 Integrar Componentes de software:** El proyecto APT integra múltiples APIs y servicios como Google Maps para geolocalización y pasarelas de pago para transacciones.

**Relación del proyecto-intereses profesionales**

Nuestro proyecto APT tiene relación con intereses **desarrollo móvil**.

**Desarrollo Móvil**

Desarrollado en **React Native**, el proyecto APT creará una buena experiencia en múltiples plataformas. La integración de APIs como **Google Maps** y **pasarelas de pago** ampliará la funcionalidad, mientras que el enfoque en UX asegurará una interfaz intuitiva para usuarios y distribuidores.

# **Factibilidad**

La aplicación es factible de llevar a cabo, aunque con ciertas dificultades. El proyecto APT es viable gracias a herramientas como React Native, el cual permite desarrollar aplicaciones híbridas (iOS y Android). Además, el proyecto se enfocará mucho en las funcionalidades principales, para así pulir al máximo estas.

**Propuesta metodológica**

**1. Planificación Inicial**

**1.1. Definición de Objetivos**

* **Objetivos Generales:** Desarrollar una aplicación móvil en React Native que permita a los usuarios solicitar gas a distribuidores independientes, integrando funcionalidades de geolocalización, pago en línea, y gestión de pedidos.
* **Objetivos Específicos:**
  + Crear perfiles diferenciados para usuarios, distribuidores y administradores.
  + Implementar un sistema de pedidos en tiempo real con seguimiento.
  + Integrar el API de Google Maps para geolocalización y cálculo de rutas.
  + Incorporar pasarelas de pago seguras y manejo de errores.

**1.2. Toma de Requerimientos**

* **Reuniones con Stakeholders:** Entrevistas con futuros usuarios, distribuidores y administradores para entender sus necesidades y expectativas.
* **Análisis Competitivo:** Evaluación de aplicaciones similares en el mercado para identificar fortalezas y debilidades.

**2. Ciclo de Desarrollo**

**2.1. Backlog del Producto**

* **Creación del Backlog:** Lista priorizada de todas las funcionalidades, mejoras y correcciones necesarias para el producto.

**2.2. Planificación de Sprints**

* **Duración de los Sprints:** Sprints de 1 a 3 semanas.
* **Planificación del Sprint:** Selección de las historias de usuario que se trabajarán en cada sprint basándose en su prioridad y en la capacidad del equipo.
* **Definición de Hecho:** Criterios específicos que deben cumplirse para que una historia de usuario se considere completada.

**2.3. Desarrollo e Implementación**

* **Desarrollo Incremental:** Se trabajará en la implementación de las funcionalidades de forma iterativa e incremental.
* **Revisión de Código:** Realización de revisiones de código para asegurar la calidad del desarrollo.
* **Integración Continua (CI):** Automatización de pruebas unitarias y funcionales cada vez que se realiza una nueva commit al repositorio.

**2.4. Pruebas**

* **Pruebas Unitarias:** Asegurar que cada componente funciona correctamente de manera aislada.
* **Pruebas de Integración:** Verificar que los módulos y servicios interactúan correctamente entre sí.
* **Pruebas de Usuario (Beta Testing):** Evaluación de la aplicación por un grupo reducido de usuarios para detectar posibles mejoras en la usabilidad.

**2.5. Retrospectiva del Sprint**

* **Revisión del Sprint:** Análisis del trabajo realizado, revisión de las historias completadas y evaluación de los desafíos enfrentados.
* **Mejoras Continuas:** Identificación de áreas de mejora para aplicar en los próximos sprints.

**3. Despliegue y Mantenimiento**

**3.1. Despliegue Automatizado**

* **Despliegue Gradual:** Implementación de la aplicación a un porcentaje pequeño de usuarios inicialmente, para detectar errores en producción antes de un lanzamiento completo.

**3.2. Monitoreo post-Despliegue**

* **Monitoreo en Tiempo Real:** Uso de herramientas como Firebase o Sentry para detectar errores en la aplicación una vez desplegada.

**3.3. Actualizaciones y Soporte**

* **Ciclo de Actualizaciones:** Planificación de actualizaciones regulares para mejorar la funcionalidad, seguridad y rendimiento de la aplicación.

**4. Gestión de Riesgos**

**4.1. Identificación de Riesgos**

* **Riesgos Técnicos:** Problemas relacionados con la integración de APIs, rendimiento de la aplicación, etc.
* **Riesgos Operacionales:** Disponibilidad del equipo, cambios, etc.
* **Riesgos de Seguridad:** Vulnerabilidades en la protección de datos y en la pasarela de pago.

**4.2. Mitigación de Riesgos**

* **Plan de Contingencia:** Estrategias para mitigar y resolver los riesgos identificados.

**5. Documentación**

* **Documentación Técnica:** Incluye arquitectura, APIs utilizadas, y estructura del código.
* **Manual de Usuario:** Guía para usuarios finales sobre cómo utilizar la aplicación.
* **Documentación de Despliegue:** Pasos detallados para la implementación y mantenimiento de la aplicación en entornos de producción.

**6. Cierre del Proyecto**

* **Revisión Final:** Evaluación del cumplimiento de los objetivos y entrega del producto.
* **Lecciones Aprendidas:** Análisis de lo que funcionó bien y lo que se puede mejorar para futuros proyectos.

# 

# **Plan de trabajo para el proyecto APT**

**Introducción**

El Proyecto APT consiste en el desarrollo de una aplicación móvil que permitirá a los usuarios solicitar la entrega de gas por parte de distribuidores independientes en Chile. El objetivo principal es crear una solución que optimice la interacción entre clientes y distribuidores, utilizando funcionalidades avanzadas como geolocalización, seguimiento en tiempo real, y pasarelas de pago seguras.

**2. Metodología**

Se utilizará la metodología **Scrum**, para asegurar un desarrollo ágil e iterativo. Esta metodología permitirá realizar entregas parciales del producto y ajustes según las necesidades del usuario.

**Fases del Proyecto**

**Fase 1: Planificación y Análisis (Semana 1 - Semana 2)**

1. **Definición del Proyecto:**
   * Objetivos, alcances y requerimientos del proyecto.
   * Análisis de las necesidades funcionales y no funcionales.
2. **Documentación de Requerimientos:**
   * Creación de un documento de requerimientos detallado que incluya historias de usuario, flujos de trabajo y casos de uso.
3. **Especificación Técnica:**
   * Definición de las tecnologías a utilizar:
     1. React Native (Expo),
     2. Google Maps API,
     3. pasarelas de pago,
     4. OAuth para autenticación,
     5. NativeWind para los estilos,
     6. Neon PostgreSQL database
   * Elaboración de la arquitectura del sistema (Backend, Frontend, Base de Datos).

**Fase 2: Diseño del Sistema (Semana 3 - Semana 4)**

1. **Diseño de Interfaces y UX/UI:**
   * Creación de wireframes y prototipos de las pantallas de la aplicación (perfiles de usuario, distribuidores, administrador, etc.).

**Fase 3: Desarrollo Backend y Frontend (Semana 4 - Semana 12)**

1. **Sprint 1 (Semana 4 - Semana 6):**
   * **Funcionalidad:** Sistema de autenticación (registro, login, perfil de usuario y distribuidor).
   * **Tareas:**
     + Implementación de Firebase para autenticación.
     + Creación del sistema de registro de usuarios y distribuidores.
     + Creación de perfiles (UI).
   * **Entrega:** Módulo de autenticación funcional.
2. **Sprint 2 (Semana 7 - Semana 8):**
   * **Funcionalidad:** Geolocalización y selección de distribuidores.
   * **Tareas:**
     + Integración de Google Maps API.
     + Implementación de geolocalización en tiempo real.
     + Filtrado de distribuidores cercanos.
   * **Entrega:** Funcionalidad básica de mapas y geolocalización.
3. **Sprint 3 (Semana 9 - Semana 10):**
   * **Funcionalidad:** Sistema de pedidos y seguimiento en tiempo real.
   * **Tareas:**
     + Implementación del sistema de pedidos (usuario selecciona distribuidor).
     + Seguimiento en tiempo real de distribuidores (ruta).
   * **Entrega:** Módulo de pedidos y seguimiento en mapas.
4. **Sprint 4 (Semana 11 - Semana 12):**
   * **Funcionalidad:** Pasarelas de pago y manejo de transacciones.
   * **Tareas:**
     + Integración de WebPay o MercadoPago.
     + Implementación de lógica para manejo de errores de transacción y confirmación.
   * **Entrega:** Módulo de pagos funcional.

**Fase 4: Pruebas y Calidad (Semana 13 - Semana 15)**

1. **Pruebas Unitarias y de Integración:**
   * Asegurarse de que cada módulo de la aplicación funcione de manera independiente y conjunta.
   * Pruebas de integración con las APIs (Google Maps, pasarelas de pago).
2. **Pruebas de Usuario:**
   * Testeo de la aplicación con un grupo de usuarios beta para identificar problemas de usabilidad y bugs.
3. **Ajustes Finales:**
   * Corrección de bugs y mejoras según feedback recibido en las pruebas.

**Fase 5: Despliegue (Semana 16 - Semana 17)**

1. **Despliegue en Ambiente de Producción:**
   * Configuración del entorno de producción.
   * Despliegue de la aplicación en tiendas de apps (Google Play y App Store).
2. **Monitoreo y Soporte Inicial:**
   * Implementación de herramientas de monitoreo para asegurar que no haya fallos críticos en la aplicación.

**3. Recursos**

1. **Equipo de Desarrollo:**
   * 2 Desarrolladores Frontend (React Native).
   * 2 Desarrollador Backend (Firebase, Node.js).
   * 2 Ingeniero DevOps.
2. **Herramientas:**
   * **Gestión del Proyecto:** Jira.
   * **Control de Versiones:** GitHub.
   * **Integración Continua:** GitHub Actions o Jenkins.
   * **Pruebas:** Jest (unitarias).

**4. Cronograma General**

|  |  |  |  |
| --- | --- | --- | --- |
| Fase | Duración | Fecha de Inicio | Fecha de Término |
| Planificación y Análisis | 3 semanas | Semana 1 | Semana 3 |
| Diseño del Sistema | 2 semanas | Semana 3 | Semana 4 |
| Desarrollo | 8 semanas | Semana 5 | Semana 12 |
| Pruebas y Calidad | 3 semanas | Semana 13 | Semana 15 |
| Despliegue | 2 semanas | Semana 16 | Semana 17 |
|  |  |  |  |

**Propuesta de evidencias**

**1.** **Diseño y Desarrollo de la Aplicación**

* **EV1**: **Wireframes y Mockups**: Los diseños de la interfaz de usuario (UI) desarrollados en herramientas como Figma.
* **EV2**: **Repositorio de Código**: El repositorio de código (por ejemplo, en GitHub), que debe incluir commits con descripciones claras, ramas de desarrollo, y documentación técnica (README, comentarios en el código, etc.).

**2.** **Implementación de Funcionalidades Específicas (e.g., Geolocalización, Pagos)**

* **EV1: Pruebas Unitarias**: Resultados de pruebas unitarias automatizadas que demuestran que las funciones específicas funcionan según lo previsto.
* **EV2:** **Demostración de Funcionalidad**: Capturas de pantalla o videos que muestren la implementación y el funcionamiento de las características específicas dentro de la aplicación.

**3. Integración con Servicios Externos (e.g., API de Google Maps, Gateway de Pagos)**

* **EV1**: **Documentación de Integración**: Manuales o guías que describan cómo se integraron las APIs, incluyendo detalles de autenticación, endpoints utilizados, y respuestas esperadas.
* **EV2**: **Pruebas de Integración**: Resultados de pruebas que muestren la comunicación exitosa con las APIs externas y su funcionamiento dentro de la aplicación.

**4.** **Pruebas y Validación**

* **EV1: Plan de Pruebas**: Documento detallado que describa los casos de prueba, criterios de aceptación, y escenarios cubiertos.
* **EV2**: **Resultados de Pruebas**: Reportes que muestren los resultados de las pruebas realizadas, incluyendo pruebas de carga, estrés, y usabilidad.

**5.** **Despliegue**

* **EV1**: **Guía de Despliegue**: Documentación que detalla el proceso de despliegue en el entorno de producción, incluyendo pasos, configuraciones, y procedimientos de rollback.