



**Capstone**

**Docente:** Arturo Guerra

Índice

[1. Antecedentes Personales 3](#_gz19oixy3buj)

[2. Descripción Proyecto APT 4](#_m03givrlb796)

[3. Fundamentación Proyecto APT 6](#_s8vbb1thzi6y)

[4. Objetivos 13](#_jivuwagapis3)

[5. Metodología 14](#_otv197f9dai9)

[6. Evidencias 16](#_fet9bluceek0)

[7. Plan de Trabajo 17](#_2xvdq2v83rsj)

[8. Carta Gantt 21](#_6neavssujyej)

# 

1. **PARTE I**

| 1. Antecedentes Personales |
| --- |
| A continuación, se presenta una tabla en la que debes completar la información solicitada. |

| Nombre estudiante | **Esteban Díaz Contreras** |
| --- | --- |
| Rut | **21.216.531-k** |
| Carrera | **Ingeniería en informática** |
| Sede | **Antonio Varas** |

| Nombre estudiante | **Diego Gieminiani González** |
| --- | --- |
| Rut | **18.769.251-2** |
| Carrera | **Ingeniería en informática** |
| Sede | **Antonio Varas** |

| Nombre estudiante | **Marian Moreno Ortega** |
| --- | --- |
| Rut | **26.124.470-5** |
| Carrera | **Ingeniería en informática** |
| Sede | **Antonio Varas** |

| **2. Descripción Proyecto APT** |
| --- |
| En la descripción debes señalar brevemente el nombre de tu proyecto APT y las competencias del perfil de egreso que vas a poner en práctica. Si en tu carrera están definidas las áreas de desempeño, también menciona a qué áreas de desempeño está vinculado el proyecto. |

| Nombre del proyecto | QA Station |
| --- | --- |
| Área (s) de desempeño(s) | Para el proyecto QA Station, las áreas de desempeño que abordaremos de acuerdo a tu plan de estudio son:   * Análisis y Evaluación de Soluciones Informáticas * Desarrollo de Software * Gestión de proyectos informáticos |
| Competencias | Las competencias del plan de estudio que vamos a abordar en tu Proyecto son:  **C1: Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria.**  QA Station está diseñada para automatizar las pruebas de calidad de software, promoviendo la adopción de buenas prácticas en QA. La plataforma permite a los usuarios ejecutar pruebas funcionales automatizadas de manera eficiente, asegurando que los productos cumplan con los más altos estándares de calidad. Además, integra IA para optimizar el proceso, aumentando la precisión y eficiencia en la realización de pruebas.  **C2: Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización.**  La plataforma será gestionada utilizando metodologías ágiles (Scrum). Esto permitirá gestionar eficientemente el desarrollo del proyecto, priorizar tareas, y ofrecer soluciones que optimicen los recursos de QA.  **C3: Construir modelos de datos para soportar los requerimientos de la organización de acuerdo a un diseño definido y escalable en el tiempo.**  La plataforma requerirá la creación de un sistema que gestione de manera eficiente los casos de prueba y los resultados de las pruebas automatizadas. Este modelo de datos debe ser escalable para soportar un número creciente de usuarios y casos de prueba, alineándose con las necesidades futuras del proyecto.  **C4: Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos.**  La creación de QA Station implica el desarrollo de una plataforma que automatiza y sistematiza las pruebas de calidad. Esto incluye la integración de IA para la generación de casos de prueba y la automatización de pruebas funcionales con Selenium, asegurando la consistencia y eficiencia en los procesos de QA. |

| 3. Fundamentación Proyecto APT |
| --- |
| A continuación, se presentan distintos campos que debes completar con la información solicitada. Esta sección busca que describas en detalle tu proyecto y justifiques su relevancia y pertinencia. |

| Relevancia del proyecto APT | El proyecto QA Station surge en respuesta a la necesidad de implementar y gestionar procesos de aseguramiento de la calidad (QA) profesionales y personas en formación con recursos limitados. En la industria del software, la automatización de pruebas es esencial para garantizar productos de alta calidad, pero muchas personas carecen de las herramientas o el conocimiento adecuado para implementar este tipo de soluciones. QA Station busca resolver esta problemática proporcionando una plataforma accesible que facilite la creación y ejecución de pruebas funcionales automatizadas, utilizando herramientas avanzadas como Selenium y funcionalidades asistidas por IA.  **Problemática:** Uno de los problemas más recurrentes en la industria del software es la falta de gestión en los procesos de QA. Esto puede generar productos con errores, retrasos en el ciclo de desarrollo y, en última instancia, una experiencia insatisfactoria para los usuarios. Las pruebas manuales son costosas y lentas, mientras que las soluciones de automatización disponibles en el mercado a menudo requieren conocimientos técnicos avanzados, lo que crea una barrera para muchos profesionales. *QA Station* aborda esta problemática al proporcionar una plataforma que simplifica el proceso de automatización, permitiendo a los usuarios generar casos de prueba y ejecutar pruebas funcionales sin la necesidad de dominar herramientas más complejas.  **Contexto:** El proyecto se sitúa en el contexto de la industria del software en Chile, donde muchos profesionales y personas en formación están adoptando tecnologías modernas, pero enfrentan desafíos al implementar buenas prácticas de QA. En un mercado donde la demanda de productos digitales crece exponencialmente, contar con soluciones accesibles que ayuden a mejorar la calidad de sus productos es esencial para mantener la competitividad. Además, el proyecto también impacta en el sector educativo, ayudando a los estudiantes y profesionales en formación a adquirir experiencia práctica en QA automatizado.  **Impacto:** *QA Station* está diseñado para impactar a audiencias principales:   1. **Profesionales en formación y estudiantes:** La plataforma también está orientada a quienes están aprendiendo sobre QA. Al proporcionar una herramienta accesible y fácil de usar, permite que estos profesionales adquieran habilidades prácticas en un área fundamental para la industria del software.   **Aporte al Campo Laboral:** La automatización de pruebas es una de las áreas más demandadas en la ingeniería de software moderna. QA Station no solo busca solucionar una necesidad práctica en el campo del QA, sino que también contribuye a la profesionalización del sector, facilitando el acceso a herramientas que normalmente están reservadas para grandes empresas con recursos técnicos avanzados. El proyecto también ofrece una solución innovadora al incorporar IA para asistir en la creación de casos de prueba, lo que aumenta la eficiencia y democratiza los procesos sin requerir conocimientos especializados en programación y herramientas específicas. |
| --- | --- |
| Descripción del Proyecto APT | El proyecto QA Station consiste en el desarrollo de una plataforma web que facilite la automatización y gestión de pruebas de calidad de software (QA), dirigida a profesionales en formación de desarrollo de software. Este sistema proporcionará a los usuarios una solución accesible y simplificada para ejecutar pruebas automatizadas, integrando capacidades de inteligencia artificial (IA) para mejorar la eficiencia y simplificar el proceso de creación de casos de prueba.  Como requerimientos de alto nivel, se espera que la plataforma cumpla con las siguientes funciones:  **Uso interno de la plataforma (para desarrolladores):**   1. **Módulo de Documentación:**    * Generación automática de casos de prueba a partir de indicaciones dadas por el usuario en lenguaje natural, gracias al uso de IA.    * Almacenamiento y gestión de casos de prueba, permitiendo su posterior consulta y reutilización.    * Guías detalladas sobre cómo utilizar las funcionalidades de la plataforma y las mejores prácticas de QA. 2. **Módulo de Generación de Casos de Prueba con IA:**    * Generación automática de casos de prueba al subir documentos como especificaciones de funcionalidades del sistema en formato PDF o word.    * La IA analiza el contenido del documento y genera casos de prueba basados en los detalles proporcionados, asegurando la cobertura de las funcionalidades descritas.    * Los casos generados se almacenan para su consulta, edición y reutilización futura para el módulo de pruebas automatizadas.. 3. **Módulo de Ejecución de Pruebas Automatizadas:**    * Automatización de pruebas funcionales basadas en tags HTML utilizando Selenium. Los usuarios solo necesitan ingresar la URL y completar los inputs necesarios, mientras que el sistema ejecutará automáticamente las pruebas.    * Identificación y validación de elementos HTML mediante IA, lo que simplifica el proceso de prueba para usuarios con poca o ninguna experiencia en programación.    * Generación de vistas que muestran si las pruebas han sido exitosas (OK) o han fallado (No OK), con un resumen de los flujos de prueba.   **Uso público:**   1. **Inscripción en la plataforma:** Profesionales en formación pueden registrarse y acceder a la plataforma para generar y ejecutar pruebas. 2. **Acceso a Documentación:** Los usuarios tendrán acceso a las guías de mejores prácticas y documentación técnica generada por la plataforma.   **Características técnicas del sistema:**   * La plataforma estará diseñada para ser fácil de usar, con un enfoque en la usabilidad para que los usuarios, sin conocimientos técnicos avanzados, puedan gestionar el sistema y ejecutar pruebas sin dificultad.   **Automatización Accesible:** La incorporación de IA y Selenium democratiza el acceso a herramientas avanzadas de QA, permitiendo que los usuarios adopten estas prácticas sin necesidad de conocimientos especializados. La plataforma facilitará el aprendizaje y simplificación de los procesos de QA, mejorando la calidad y consistencia del software desarrollado. |
| Pertinencia del proyecto con el perfil de egreso | El Proyecto se alinea estrechamente con el perfil de egreso de la carrera, que se centra en la formación de profesionales capaces de desarrollar, gestionar y asegurar la calidad de soluciones informáticas, aplicando buenas prácticas y utilizando diversas tecnologías. Este proyecto no solo aplica las competencias técnicas necesarias para abordar problemas reales en el ámbito del desarrollo de software, sino que también contribuye al fortalecimiento de la calidad y la gestión de los procesos de calidad (QA), un área importante en la industria.  **Relación con el Perfil de Egreso:** El proyecto refleja las capacidades que un egresado de la carrera debe poseer, particularmente en términos de:   * **Desarrollo y Automatización de Soluciones de Software:** El desarrollo de una plataforma que automatiza pruebas funcionales de QA es una aplicación directa de las habilidades de programación, diseño de software y uso de herramientas avanzadas, todas ellas fundamentales en el perfil de egreso. * **Gestión de Proyectos Informáticos:** La planificación, ejecución y control de este proyecto demuestran competencias en la gestión de proyectos tecnológicos, asegurando que se cumplan los objetivos dentro de los plazos y recursos disponibles. * **Aplicación de Buenas Prácticas:** A través del diseño y la implementación de la plataforma, se pone en práctica la competencia de aplicar buenas prácticas de la industria en la certificación de productos y procesos, asegurando que el software desarrollado cumpla con los estándares de calidad.   **Necesidad de las Competencias Seleccionadas:** Las competencias seleccionadas son esenciales para resolver la problemática que se aborda en el Proyecto:   1. **C1: Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria.**   **Relevancia en el Proyecto:** Esta competencia es clave para garantizar que las pruebas automatizadas que ofrece la plataforma sean efectivas y se alineen con las mejores prácticas de la industria, lo cual es esencial para asegurar la calidad del software.   1. **C2: Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización.**   **Relevancia en el Proyecto:** La gestión adecuada del proyecto, desde la concepción hasta la implementación, requiere una toma de decisiones informada que optimice los recursos y tiempos disponibles, asegurando el éxito del proyecto.   1. **C4: Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos.**   **Relevancia en el Proyecto:** La competencia de desarrollar soluciones de software sistematizadas es crucial para la creación de una plataforma que no solo automatiza tareas complejas, sino que también asegura que estas tareas se realicen de manera consistente y repetible, logrando los objetivos de calidad planteados. |
| Relación con los intereses profesionales | El Proyecto está alineado con nuestros intereses profesionales, los cuales se centran en la especialización en aseguramiento de la calidad del software (QA). Junto con mi equipo, hemos dedicado nuestros esfuerzos a profesionalizarse en este campo, reconociendo la importancia de establecer y mantener estándares de calidad en el desarrollo de software.  **Intereses Profesionales:** El interés profesional radica en contribuir al avance del QA en la industria del software, asegurando que las aplicaciones y sistemas desarrollados cumplan con los más altos estándares de calidad. Este interés abarca tanto el QA manual, donde se requiere un enfoque detallado y meticuloso, como el QA automático, que permite escalar y sistematizar los procesos de prueba para lograr una mayor eficiencia y consistencia.  **Aspectos Reflejados en el Proyecto:** El Proyecto refleja directamente estos intereses profesionales al enfocarse en la automatización de pruebas de software. La creación de una plataforma que no solo facilita el manejo sino que también la gestión de estos procesos, asegura que los productos desarrollados mantengan un alto nivel de calidad. Este proyecto es una extensión natural de nuestro interés por mejorar y optimizar las prácticas de QA, permitiendo a profesionales en formación y futuramente a equipos pequeños de trabajo que adopten estas prácticas de manera más accesible y eficiente.  **Contribución al Desarrollo Profesional:** Realizar este Proyecto contribuirá significativamente a nuestro desarrollo profesional al permitir aplicar y expandir nuestros conocimientos en QA automático, una área que es cada vez más crucial en la industria del software. Además, este proyecto nos permitirá desarrollar habilidades clave en la gestión de proyectos tecnológicos, el desarrollo de software, y la implementación de buenas prácticas, todas ellas fundamentales para avanzar en nuestra carrera. El éxito en este proyecto también demostrará nuestra capacidad para identificar y abordar problemas reales en la industria, aportando soluciones innovadoras que tienen un impacto tangible en la calidad del software. |
| Factibilidad de desarrollo del Proyecto APT | Consideramos que la factibilidad de desarrollo de este proyecto es completamente viable, basándonos en los siguientes aspectos clave:   1. Desarrollar el Proyecto es posible y factible dentro del marco del semestre académico, tomando en cuenta los recursos disponibles, el tiempo asignado, y la naturaleza del proyecto como un MVP (Minimum Viable Product). 2. **Duración del Semestre y Horas Asignadas:** El proyecto está diseñado para ser completado dentro de la duración del semestre, aprovechando las horas asignadas a la asignatura. La planificación del proyecto se ha ajustado para que los hitos principales, como el desarrollo del módulo de documentación y el módulo de ejecución de pruebas, se completen de manera secuencial y dentro del tiempo disponible. Este enfoque permite una implementación efectiva del MVP, asegurando que se logre un producto funcional y utilizable en el tiempo estipulado. 3. **Materiales Requeridos:** Los materiales necesarios para desarrollar el proyecto son accesibles y se encuentran dentro de las capacidades del equipo. Esto incluye herramientas de desarrollo como Python y Selenium para la automatización de pruebas, Django para el backend y frontend. Además, el equipo cuenta con los conocimientos técnicos necesarios para implementar estas tecnologías, lo que minimiza la necesidad de adquirir recursos adicionales. 4. **Factores Externos que Facilitan el Desarrollo:** Un factor que facilita el desarrollo del proyecto es el acceso a un entorno de desarrollo adecuado, con todas las herramientas y recursos necesarios ya configurados y disponibles. Además, la experiencia previa del equipo en QA y desarrollo de software es un facilitador clave, permitiendo que se aborden las tareas del proyecto de manera eficiente y efectiva. 5. **Factores Externos que Podrían Dificultar el Desarrollo y Soluciones:** Existen factores externos que podrían representar desafíos, como posibles interrupciones en el acceso a las herramientas de desarrollo debido a problemas técnicos o limitaciones de tiempo no previstas. 6. **MVP (Minimum Viable Product):** Al tratarse de un MVP, el enfoque del proyecto está en desarrollar una versión funcional que incluya las características esenciales. Esto asegura que el proyecto sea manejable dentro del tiempo disponible y que cualquier iteración o mejora futura se base en una plataforma sólida. La creación de un MVP permite que el proyecto se concentre en las funcionalidades más críticas, asegurando que se entregue un producto que cumpla con los objetivos principales y que pueda ser expandido en el futuro si es necesario. |

1. **PARTE II**

| 4. Objetivos |
| --- |
| En este apartado debes definir objetivos generales y específicos del Proyecto APT. Es importante aclarar que los objetivos se deben plantear en forma clara, concisa y sin dar mayores explicaciones, es decir, deben entenderse por sí solos. Se sugiere redactarlos utilizando un verbo en infinitivo, pues ello obliga a precisar acciones concretas. |

| Objetivo general | El proyecto busca superar los desafíos que enfrentan tanto personas en formación como profesionales al ejecutar y gestionar pruebas de calidad de software (QA) de manera eficiente y accesible. Su objetivo es simplificar el proceso, eliminando la necesidad de conocimientos técnicos avanzados, mientras se asegura la calidad del software y se promueve la adopción de buenas prácticas de QA. |
| --- | --- |
| Objetivos específicos | 1. **Facilitar la creación de casos de prueba para usuarios con poca experiencia técnica**, aprovechando la inteligencia artificial (IA) para generar automáticamente estos casos a partir de descripciones en lenguaje natural. 2. **Simplificar la ejecución de pruebas funcionales** automatizando la identificación de elementos HTML y la ejecución de los tests, utilizando herramientas como Selenium, además de la IA. 3. **Mejorar la accesibilidad y usabilidad del sistema**, asegurando que incluso usuarios sin un conocimiento avanzado en QA puedan ejecutar pruebas automatizadas de manera intuitiva y eficiente. 4. **Gestionar el desarrollo del proyecto de forma eficiente** a través de metodologías ágiles, como Scrum, asegurando una organización óptima, planificación efectiva y entregas continuas de valor durante todo el ciclo de desarrollo. |

| 5. Metodología |
| --- |
| En el siguiente apartado deberás describir la metodología, propia de tu disciplina, que utilizarás para resolver el proyecto APT antes descrito, incluyendo las etapas y métodos de trabajo. |

| Descripción de la Metodología |
| --- |
| Para abordar la problemática identificada y cumplir con los objetivos del proyecto QA Station, se aplicará una metodología ágil para gestión de proyecto, enfocada en entregar valor constante a través de la automatización de pruebas de calidad de software. La metodología ágil para gestión de proyecto será Scrum, dado que facilita la organización del trabajo en ciclos iterativos (sprints), lo que permite flexibilidad, adaptación a los cambios y esto nos ayudará a generar valor en cada entrega.  Inicialmente, se realizará un análisis detallado de las pruebas que se realizarán para cumplir el desafío de automatizar pruebas, utilizando herramientas de mapeo como mapas: de actores y mapas mentales para definir los roles, necesidades y expectativas de los usuarios. Esta etapa también incluirá la declaración de la visión del proyecto, que se estructurará en cuatro pilares clave: viabilidad, beneficios, riesgos y alcance. Se validará este análisis mediante mapas de impacto, asegurando que las soluciones propuestas cubran todas las áreas críticas.  Posteriormente, se definirán épicas e historias de usuario para las distintas funcionalidades claves de la plataforma, como la generación automática de casos de prueba mediante IA y la ejecución de pruebas funcionales automatizadas con Selenium. Estas historias se estructurarán en historias de usuarios que guiará el desarrollo. El Product Backlog será priorizado y posteriormente estimado para asegurar que las funcionalidades más importantes sean abordadas primero.  El desarrollo del aplicativo se realizará en ciclos iterativos e incrementales mediante sprints. El ciclo inicial, denominado "Sprint Cero", se dedicará a la configuración de la base de datos, la interfaz tecnológica y la configuración inicial de la plataforma. Luego, se desarrollarán cuatro prints adicionales, cada uno de los cuales incluirá:   * Sprint Planning para definir las tareas a realizar. * Daily Scrum para visualizar el progreso diario. * Burndown Chart y scrum board para controlar el ritmo de trabajo. * Sprint Review y Retrospective para evaluar el resultado y buscar e implementar mejoras al producto o procesos.   Durante cada sprint, el equipo gestionará las tareas y el progreso utilizando Jira, donde se registrarán impedimentos, se actualizará el backlog del producto, se documentarán las soluciones entregadas y se actualizará el tablero Jira “”Scrum board”.  Finalmente, se llevará a cabo un proceso de validación y verificación para asegurar que la plataforma cumpla con las historias de usuarios definidas y los objetivos de automatización. Al finalizar el proyecto, se llevará a cabo una revisión del producto, seguida de una retrospectiva final. Posteriormente, se procederá con el cierre oficial del proyecto  **Roles y responsabilidades:** El proyecto se desarrollará en equipo, y los roles serán distribuidos de la siguiente manera:   * Scrum Master: Encargado de facilitar el proceso ágil, eliminando obstáculos y asegurando que el equipo pueda trabajar eficientemente. * Product Owner: Responsable de gestionar el backlog del producto y priorizar las funcionalidades que se desarrollarán en cada sprint. * Developers: Encargados del desarrollo técnico de la plataforma, incluyendo la integración de Selenium, la implementación de la IA y la gestión del backend y frontend. * Ingeniero QA: Encargado de realizar las pruebas y asegurar la calidad del producto entregado. |

| 6. Evidencias |
| --- |
| A continuación, describe qué evidencias serán evaluadas en el informe de avance y en el informe final de tu proyecto APT. Estas evidencias deben ser acordadas con tu docente. Se entenderá por evidencia los productos que se desarrollen durante el proyecto y cuyo propósito sea visibilizar o documentar cómo se ha implementado el trabajo. |

| **Tipo de evidencia (interna o final).** | **Nombre de la evidencia** | **Descripción** | **Justificación** |
| --- | --- | --- | --- |
| Documentación Técnica y Planificación | Documentación General del Proyecto | Esta evidencia agrupa los documentos clave que incluyen documentación como el Plan del Proyecto, la Definición del Proyecto, y otros. Todos estos documentos estarán almacenados en el mismo **repositorio de GitHub** denominado **QA-Station** para visibilizar cómo se ha implementado el trabajo. | Estos documentos son fundamentales para guiar el desarrollo, asegurando que se sigan los objetivos definidos y se mantenga un seguimiento claro del progreso. |
| Reflexión y Autoevaluación por Fases | Reflexión y Autoevaluación del Proyecto | Documentos que detallan la autoevaluación del equipo e individual, reflexionando sobre el desarrollo de competencias y aprendizaje obtenido. Todos estos documentos estarán almacenados en el mismo **repositorio de GitHub** denominado **QA-Station** para visibilizar cómo se ha implementado el trabajo. | Permite reflexionar sobre el desarrollo de habilidades, aprendizajes y mejoras en futuras iteraciones del proyecto. |
| Gestión de Proyecto | Gestión de Tareas en Jira | Uso de Jira para la organización de sprints y seguimiento de las tareas relacionadas con el desarrollo del proyecto. | Muestra cómo se gestionó el trabajo y el avance del equipo, permitiendo el seguimiento del progreso a lo largo de los sprints. |
| Diseño de Interfaz | Prototipos de Interfaz en Figma | Prototipos diseñados en Figma, que muestran la evolución de la interfaz, desde wireframes hasta la versión final. | Refleja cómo se diseñó y estructuró la interfaz, documentando el proceso iterativo de diseño. |
| Control de Versiones | Código Fuente en GitHub | Repositorio en GitHub llamado **QA\_STATION** donde se aloja el código fuente del proyecto con control de versiones. | Visibiliza el desarrollo del código y facilita la revisión colaborativa, asegurando control de versiones. |

| 7. Plan de Trabajo |
| --- |
| En la siguiente tabla define la planificación de tu Proyecto APT de acuerdo a lo requerido. |

| **Plan de Trabajo Proyecto APT** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Competencia o unidades de competencias | Nombre de Actividades/Tareas | Descripción Actividades/Tareas | Recursos | Duración de la actividad | Responsable[[1]](#footnote-0) | Observaciones |
| Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización. | -Clarificación de requisitos.  -Definición de alcance.  -Reducción de riesgos.  -Establecimiento de criterios de aceptación. | -Clarificación de Requisitos: Definir y aclarar los requisitos del proyecto para asegurar una comprensión común entre el equipo y los stakeholders.  -Definición de Alcance: Delimitar el alcance del proyecto, especificando qué se incluirá y qué no, para gestionar expectativas y evitar cambios innecesarios.  -Reducción de Riesgos: Identificar y mitigar riesgos potenciales que puedan impactar el éxito del proyecto, minimizando problemas futuros.  -Establecimiento de Criterios de Aceptación: Definir los criterios específicos que deben cumplirse para considerar que una tarea o funcionalidad está completa y es aceptable. | **Clarificación de Requisitos**: -Product Owner: Para definir y aclarar requisitos. -Equipo de Desarrollo: Para entender los requisitos.  **Definición de Alcance**: -Analista de Negocios: Para delimitar el alcance. -Product Owner: Para validar el alcance.  **Reducción de Riesgos**: -Scrum Master: Para identificar y mitigar riesgos. -Equipo de Desarrollo: Para señalar posibles riesgos técnicos.  **Establecimiento de Criterios de Aceptación**: -Product Owner: Para definir los criterios de aceptación. -Equipo de Desarrollo: Para implementar y verificar los criterios. | 2 semanas. | Marian Moreno | El alcance del proyecto no estaba muy claro |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | -Sprint Planning  -Desarrollo de funcionalidades  -Implementación de pruebas automatizadas. | Sprint Planning: Planificar el trabajo del sprint, seleccionando y detallando las tareas y funcionalidades a desarrollar.  Desarrollo de Funcionalidades: Implementar y codificar las funcionalidades del software según los requisitos y prioridades del sprint.  Implementación de Pruebas Automatizadas: Crear y ejecutar pruebas automatizadas para verificar que el software funcione correctamente y cumpla con los estándares establecidos. | **Sprint Planning:** -Product Owner: Para definir y priorizar las tareas y funcionalidades. -Scrum Master: Para facilitar la planificación del sprint. -Equipo de Desarrollo: Para estimar y planificar el trabajo.  **Desarrollo de Funcionalidades:** -Desarrolladores: Para codificar e implementar las funcionalidades. -Herramientas de Desarrollo: Como IDEs, sistemas de control de versiones (Git), y entornos de desarrollo. -Implementación de Pruebas Automatizadas:  **Ingenieros de QA:** Para diseñar y desarrollar las pruebas automatizadas.  **Herramientas de Pruebas** Automatizadas: Como Selenium.  **Entorno de Pruebas:** Para ejecutar y validar las pruebas automatizadas. | 2 semanas. | Esteban Diaz/ Diego Gieminiani | Algunas de las dificultades que tuvimos fue elegir cual de las inteligencias artificiales era mejor para nuestro proyecto. |
| Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria. | -Análisis del requerimiento  -Diseño del plan de pruebas.  -Ejecución de pruebas.  -Reporte de pruebas. | Análisis del Requerimiento: Revisar y entender los requisitos para definir las pruebas necesarias.  Diseño del Plan de Pruebas: Crear un plan detallado de pruebas que especifique cómo se validarán los requisitos del producto y los procesos.  Ejecución de Pruebas: Realizar las pruebas según el plan para verificar que el producto y los procesos cumplan con los estándares de certificación.  Reporte de Pruebas: Documentar y comunicar los resultados de las pruebas, incluyendo cualquier problema o desviación encontrada. | **Análisis del Requerimiento:** -Analista de Requisitos: Para revisar y comprender los requisitos. -Documentación del Proyecto: Para acceder a los requisitos y especificaciones.  **Diseño del Plan de Pruebas:** -Ingeniero de QA: Para diseñar el plan de pruebas detallado. -Herramientas de Gestión de Pruebas: Como TestRail o Zephyr, para documentar el plan de pruebas.  **Ejecución de Pruebas:** -Ingeniero de QA: Para llevar a cabo las pruebas. -Entorno de Pruebas: Para ejecutar las pruebas en condiciones controladas. -Herramientas de pruebas: Como Selenium, JIRA, o herramientas específicas de prueba.  **Reporte de Pruebas**: -Ingeniero QA: Para documentar y comunicar los resultados. Herramientas de Reporte: Como JIRA o sistemas de gestión de calidad, para generar informes de resultados y defectos. | 2 semanas. | Marian Moreno/ Esteban DIaz | Al momento de ejecutar las pruebas los ambientes de certificación debe estar homologados |

| 8. Carta Gantt |
| --- |
| Busca un formato de Carta Gantt que te acomode y organiza en este las actividades planificadas en el punto anterior considerando el periodo asignado para el desarrollo de tu Proyecto APT. Debes mantener la temporalidad del periodo académico en el desarrollo de las tres fases que contempla la Asignatura de Portafolio de Título. |

**La planificación debe ser en 5 sprints**

| **Actividad** | **Fase 1** | | | | **Fase 2** | | | | **Fase 3** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S 1** | **S2** | **S 3** | **S 4** | **S 5** | **S 6** | **S 7** | **S 8** | | **S 9** | **S 10** |
| Inteligencia Artificial para Generación de Casos de Prueba y Soporte en Automatización | **x** | **x** |  |  |  |  |  |  | |  |  |
| Ejecución de pruebas |  |  | **x** | **x** |  |  |  |  | |  |  |
| Documentación |  |  |  |  | **x** | **x** |  |  | |  |  |
| Administración de usuarios |  |  |  |  | **x** | **x** |  |  | |  |  |
| Login de Usuarios |  |  |  |  |  |  | **x** | **x** | |  |  |
| Cierre de sesión |  |  |  |  |  |  |  |  | | **x** | **x** |

1. En caso de que el Proyecto APT sea grupal, en esta columna deben indicar el nombre de los responsables de cada tarea o actividad. Esto posteriormente permitirá diferenciar la evaluación por cada integrante. [↑](#footnote-ref-0)