

| **1. Resumen avance Proyecto APT** |
| --- |
| A continuación, encontrarás distintos campos que deberás completar con la información solicitada. |

| Resumen de avance proyecto APT | En QA Station, nuestro objetivo es cambiar la manera en que se gestionan y ejecutan pruebas funcionales, brindando un sistema que haga más accesible la creación y ejecución de casos de prueba, especialmente para usuarios con poca experiencia en pruebas automatizadas.  Hemos logrado un avance significativo al simplificar la ejecución de pruebas funcionales mediante la automatización de la identificación de elementos HTML y la ejecución de pruebas utilizando herramientas como Selenium. Esto permite a los usuarios enfocarse en el análisis de resultados sin preocuparse por la configuración técnica detallada.  Además, estamos facilitando la creación de casos de prueba al aprovechar la inteligencia artificial para generar automáticamente estos casos a partir de descripciones en lenguaje natural. Gracias a esta funcionalidad, los usuarios pueden transformar especificaciones y requerimientos directamente en pruebas automatizadas sin necesidad de escribir código, haciendo el proceso más rápido y accesible. |
| --- | --- |
| Objetivos | **Objetivo general:**   Resolver la dificultad que enfrentan equipos pequeños y profesionales en formación al ejecutar y gestionar pruebas de calidad de software (QA) de manera eficiente y accesible, sin requerir conocimientos técnicos, avanzados, asegurando la calidad del software.  **Objetivos específicos:**   1. **Simplificar la ejecución de pruebas funcionales** automatizando la identificación de elementos HTML y la ejecución de los tests, utilizando herramientas como Selenium. 2. **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. |
| Metodología | La metodología Scrum es ideal para alcanzar los objetivos de **QA Station** por varias razones como:   * **Flexibilidad**: Scrum nos permite adaptarnos a medida que surgían nuevas ideas o desafíos. Esto es especialmente importante para un proyecto como **QA Station**, donde teníamos que integrar tecnologías complejas como IA y Selenium. A través de iteraciones constantes, podemos entender a ajustar los requisitos y mejorar las funcionalidades de forma continua, manteniendo el proyecto y alineando con los requisitos que tenemos. * **Entrega continua de valor**: Al trabajar en ciclos cortos **(sprints)**, entregamos pequeñas partes del proyecto constantemente. Esto significaba que las funcionalidades clave, como la ejecución de pruebas (Selenium) y la generación de casos de prueba, están en una constante revisión y probadas desde el principio hasta el. De esta forma, evitamos acumular problemas y reducimos riesgos, logrando validar el progreso en cada etapa. * **Enfoque en la calidad**: Cada vez que finalizamos un sprint, hacemos pruebas para asegurarnos de que el trabajo cumple con los estándares de calidad. Esto nos permite mejorar la plataforma y asegurarnos de que todo funcione de manera óptima antes de pasar a la siguiente fase. * **Reducción de riesgos**: Al revisar el proyecto después de cada sprint, podíamos detectar problemas o riesgos rápidamente y solucionarlos antes de que se convirtieran en algo mayor. Esto nos dio la tranquilidad de que, paso a paso, estábamos construyendo un producto sólido que cumplía con los objetivos iniciales. |
| Evidencias de avances | **Documentación técnica, planificación, reflexión y autoevaluación por Fases.** Esta evidencia agrupa los documentos clave como el Plan del Proyecto, la Definición del Proyecto y otros. Estos documentos permiten guiar el desarrollo, asegurando que los objetivos definidos se cumplan y que se mantenga un seguimiento claro del progreso. También, documentos que detallan la autoevaluación del equipo e individual. Toda la documentación estará almacenada en el repositorio de GitHub.  **Enlace al repositorio de GitHub QA-STATION**:  <https://github.com/mmorenoo/QA-Station> **Gestión de Proyecto (Jira)** Uso de Jira para la organización de los sprints y el seguimiento de las tareas relacionadas con el desarrollo del proyecto. Esta evidencia muestra cómo se gestionó el trabajo y el avance del equipo, permitiendo el seguimiento del progreso a lo largo de los sprints.  **Enlace a Jira**:  <https://qastation.atlassian.net/jira/software/projects/QS/boards/2/timeline> **Diseño de Interfaz (Figma)** Prototipos diseñados en Figma que muestran la evolución de la interfaz, desde wireframes hasta la versión final. Este proceso refleja cómo se diseñó y estructuró la interfaz, documentando las decisiones de diseño durante el proyecto.  **Enlace a Figma**:  <https://www.figma.com/design/UKUUINQV6jow903dDY3DhK/QA-Station?node-id=0-1&node-type=canvas&t=SAE6OiymEaduLxWJ-0> **Control de Versiones (GitHub)** El código fuente del proyecto está almacenado en un repositorio de GitHub llamado **QA\_STATION**, con control de versiones que permite revisar el desarrollo del código, facilitando la colaboración y el manejo de versiones del proyecto.  **Enlace al repositorio de GitHub QA\_STATION** <https://github.com/DiegoGieminiani/QA_STATION> |

| **Plan de Trabajo** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Competencia o unidades de competencias | Actividades | Recursos | Duración de la actividad | Responsable[[1]](#footnote-0) | Observaciones | Estado de avance | Ajustes |
| 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**: -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.*.* | 1 mes | Marian Moreno | Lo más difícil de completar fue la clarificación de requisitos*.* | Clarificación de Requisitos: **Completado**  Definición de Alcance: **Completado**  Reducción de Riesgos: **Completado**  Establecimiento de Criterios de Aceptación:  **Completado** | -Se ajustan los requisitos funcionales  -Se define mejor el alcance del sistema. |
| 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 e IA. | **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 meses | Esteban Díaz | Lo más difícil fue el desarrollo de las diferentes funcionalidades. | Sprint Planning: **Completado**  Desarrollo de Funcionalidades: **En progreso**  Análisis del Requerimiento **Completado**  Ejecución de pruebas: **En progreso** | -Se ajusta el alcance de los tipos de prueba.  -Incluimos IA en 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:** -Analista de Requisitos: Para revisar y comprender los requisitos. -Documentación del Proyecto: Para acceder a los requisitos y especificaciones.  **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 Meses | Diego Gieminiani | Lo más difícil fue la ejecución de pruebas automatizadas. | Análisis del Requerimiento**: Completado**  Ejecución de Pruebas: **En progreso**.  Reporte de Prueba**s**:**En progreso** | -Se ajusta el reporte de pruebas. |

| **3. Ajustes a partir del monitoreo** |
| --- |
| Profundiza en las observaciones de tu plan de trabajo. Analiza las actividades planificadas y señala qué aspectos facilitaron u obstaculizaron la ejecución del plan. Plantea cómo abordaste y/o abordarás los obstáculos. Por último, señala los ajustes que realizaste al plan de trabajo a partir de este análisis. |

| Factores que han facilitado y/o dificultado el desarrollo de mi plan de trabajo:  Uno de los factores que han facilitado el desarrollo de nuestro proyecto ha sido la amplia documentación que existe sobre Django. Esta herramienta cuenta con una comunidad activa y varios recursos en línea que nos han permitido avanzar rápidamente en la implementación de diversas funcionalidades. Además, la buena comunicación dentro del equipo ha sido clave para la coordinación eficiente de las tareas, lo que ha facilitado el progreso continuo del proyecto.  Por otro lado, hemos tenido algunas dificultades, principalmente relacionadas con el desconocimiento inicial sobre el uso de Selenium. Para superar esta barrera, recurrimos a tutoriales y a la consulta de la documentación oficial para entender su funcionamiento y lograr integrar correctamente el núcleo del proyecto. Asimismo, tuvimos que investigar cómo conectar nuestro sistema con la API de OpenAI para aprovechar sus capacidades, lo que implicó un aprendizaje adicional. Para enfrentar estas dificultades, nos hemos enfocado en la capacitación continua, consultando fuentes confiables y colaborando como equipo para superar los desafíos técnicos que hemos encontrado. |
| --- |

| Actividades ajustadas o eliminadas:  **Reducción de complejidad en la gestión de reportes:**Al diseñar el sistema de reportería, habíamos planeado incluir varios niveles de personalización para los informes generados. Sin embargo, decidimos simplificar esta funcionalidad y ofrecer reportes, priorizando la facilidad de uso y la implementación, evitando así la complejidad innecesaria que podía dificultar su uso para los usuarios con menos experiencia técnica.  **Eliminación de funcionalidad de múltiples tipos de usuarios (administradores)**: Habíamos planificado implementar distintos roles de usuario en el sistema, con funciones específicas para administradores. Sin embargo, al avanzar en el desarrollo y evaluar las necesidades del proyecto, decidimos eliminar esta funcionalidad. Nos dimos cuenta de que tener un solo tipo de usuario era suficiente para la plataforma QA Station, ya que nuestro enfoque principal es permitir a todos los usuarios ejecutar pruebas funcionales y generar casos de prueba de manera sencilla, sin la necesidad de una gestión compleja de roles administrativos. Esto nos permitió simplificar el desarrollo y reducir la carga de mantenimiento, enfocando los recursos en mejorar las funcionalidades clave del sistema. |
| --- |

| Actividades que no has iniciado o están retrasadas:  Actualmente, tenemos retrasadas algunas actividades clave de nuestro plan, como la implementación de las funcionalidades de login, creación de proyectos y logout. Estos retrasos se deben a que hemos priorizado el desarrollo del core del proyecto, que es la automatización de pruebas y la integración de inteligencia artificial (IA) para la generación de casos de prueba. Consideramos que enfocar nuestros esfuerzos en estas funcionalidades principales es crucial para cumplir con los objetivos de QA Station.  Sin embargo, somos conscientes de que estas funcionalidades son necesarias para el correcto funcionamiento de la plataforma. Para avanzar y evitar que estos retrasos afecten el progreso del proyecto, hemos decidido implementar una estrategia de trabajo paralelo. Esto consiste en asignar tareas específicas a distintos miembros del equipo, de manera que mientras una parte del equipo sigue trabajando en el desarrollo del core del proyecto, otro grupo se enfocará en completar estas funcionalidades pendientes. Además, estamos reevaluando nuestro cronograma para asegurar que podamos cumplir con los plazos establecidos sin comprometer la calidad del producto final. |
| --- |

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)