**Guía2. Desarrollo Proyecto APT**

**Asignatura Capstone**

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

|  |  |
| --- | --- |
| Resumen de avance proyecto APT | Hemos completado exitosamente la Fase 1 (Sprint 1) del Sistema de Gestión de Incidencias IT, cumpliendo el 100% de los objetivos específicos planteados para este hito.  Actividades realizadas:   * Configuración e implementación de la base de datos PostgreSQL con el modelo de datos completo * Desarrollo del sistema de autenticación y autorización por roles (usuario, técnico, supervisor) * Implementación del módulo completo de gestión de tickets (creación, visualización, asignación y cambio de estados) * Desarrollo del dashboard básico con métricas esenciales del sistema * Elaboración de toda la documentación técnica y de gestión del proyecto   Objetivos cumplidos:   * Sistema de autenticación funcional con control de acceso por roles * Módulo de tickets operativo con flujo completo de trabajo * Interfaz de usuario responsive e intuitiva * APIs RESTful documentadas y funcionales * Base de datos optimizada y escalable   Metodología aplicada:  Mantenemos la metodología Scrum sin cambios sustanciales, aunque realizamos ajustes menores en las estimaciones basados en la experiencia adquirida durante este primer sprint. El enfoque ágil demostró ser efectivo para gestionar los imprevistos y mantener el progreso constante.  El proyecto avanza según lo planificado, con un producto mínimo viable completamente funcional que sienta las bases sólidas para los siguientes sprints de desarrollo. |
| Objetivos | Objetivos Cumplidos en el Sprint 1:  1. Objetivo Técnico - Infraestructura Base  Configurar e implementar la base de datos PostgreSQL con el modelo de datos normalizado para tickets, usuarios y categorías.  2. Objetivo Funcional - Gestión de Tickets  Desarrollar el módulo completo de tickets que permita creación, listado, asignación y cambio de estados (Abierto → En Progreso → Resuelto).  3. Objetivo Seguridad - Control de Acceso  Implementar sistema de autenticación JWT con gestión de roles (Usuario, Técnico, Supervisor) y control de permisos.  4. Objetivo Usabilidad - Interfaz Intuitiva  Crear interfaz de usuario responsive y fácil de usar que permita a usuarios reportar incidencias en menos de 3 minutos.  5. Objetivo Calidad - Sistema Confiable  Establecer base de código mantenible con pruebas de integración y documentación técnica completa.  Objetivos Mantenidos para Sprints Futuros:   * Implementar sistema de notificaciones por email (Sprint 2) * Desarrollar reportes avanzados y métricas (Sprint 3) * Crear módulo de archivos adjuntos (Sprint 3) * Implementar integraciones con sistemas externos (Sprint 4)   No se requieren ajustes a los objetivos generales del proyecto, ya que el avance del Sprint 1 confirma que la planificación inicial es viable y adecuada para el alcance del sistema. |
| Metodología | **Metodología Implementada: Scrum**  **Enfoque principal:** Desarrollo iterativo e incremental mediante sprints de 2 semanas, con entregables funcionales al final de cada iteración.  **Ceremonias Scrum aplicadas:**   * **Sprint Planning:** Planificación detallada de 7 historias de usuario para el Sprint 1 * **Daily Meetings:** 10 reuniones diarias de 15 minutos para seguimiento * **Sprint Review:** Demostración del producto funcional al final del sprint * **Retrospectiva:** Análisis de mejora continua del proceso   **Roles del equipo:**   * **Scrum Master/Product Owner:** Diego Inostroza * **Backend Developer:** Benjamin Barrera * **Frontend Developer:** Juan Santander   **Ajustes realizados a la metodología:**  **1. Estimaciones mejoradas:** Incorporación de buffer del 20% para imprevistos técnicos en futuras planificaciones, basado en la experiencia del Sprint 1.  **2. Procesos técnicos añadidos:**   * Reuniones de sincronización técnica 2 veces por semana * Protocolo de integración frontend-backend formalizado * Revisiones de código obligatorias antes de integración   **3. Herramientas de seguimiento:**   * Excel de métricas con actualización diaria * Registro formal de impedimentos * Burndown chart para visualización de progreso   La metodología Scrum demostró ser efectiva, requiriendo solo ajustes menores en estimaciones y procesos técnicos específicos. |
| Evidencias de avance | **Evidencia 1: Repositorio de Código Funcional**  **Descripción:** Repositorio GitHub con el sistema completo implementado (backend en Node.js/Express y frontend en React.js) que incluye:   * Sistema de autenticación JWT con roles * APIs RESTful para gestión de tickets * Interfaz de usuario responsive * Base de datos PostgreSQL con datos de prueba   **Justificación:** Demuestra la funcionalidad completa del sistema y la aplicación de buenas prácticas de desarrollo software. Evidencia la correcta implementación técnica y la capacidad de entrega de productos funcionales.  **Evidencia 2: Documentación Completa del Sprint 1**  **Descripción:** 18 documentos de gestión que incluyen:   * Backlog priorizado con 15 historias de usuario * Planificación del Sprint 1 * 10 registros de Daily Meetings * Sprint Review y Retrospectiva * Mapa de actores y épicas del proyecto   **Justificación:** Evidencia la aplicación disciplinada de metodologías ágiles y gestión profesional de proyectos. Demuestra capacidad de planificación, seguimiento y mejora continua.  **Evidencia 3: Base de Datos y Scripts**  **Descripción:**   * Scripts SQL de creación de tablas (users, tickets, categories) * Datos de prueba para validación de funcionalidades * Diagrama de modelo de datos implementado   **Justificación:** Asegura la calidad mediante una base de datos bien estructurada y normalizada, aplicando principios de diseño de bases de datos relacionales.  **Resguardo de Calidad:**  La calidad se aseguró mediante:   * **Revisiones de código** entre pares para mantener estándares * **Pruebas de integración** validadas por todo el equipo * **Documentación técnica** actualizada en cada funcionalidad * **Seguimiento continuo** mediante métricas diarias de avance   **Adjuntos:** Enlace a repositorio GitHub, documentos en Google Drive, y scripts de base de datos. |

|  |
| --- |
| **2. Monitoreo del Plan de Trabajo** |
| Examina cuidadosamente tu plan de trabajo, enfocándote especialmente en la columna de estado de avance y ajustes. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Plan de Trabajo | | | | | | | |
| Competencia o unidades de competencias | Actividades | Recursos | Duración de la actividad | Responsable[[1]](#footnote-1) | Observaciones | Estado de avance | Ajustes |
| **Competencias Aplicadas en el Desarrollo del Proyecto:**  **1. Gestión de Proyectos Tecnológicos**  Planificación y seguimiento de proyectos usando metodologías ágiles (Scrum).  Gestión de recursos y tiempos en entornos de desarrollo software.  coordinación de equipos multidisciplinarios.  **2. Desarrollo de Software**  Diseño e implementación de arquitecturas de software escalables.  Programacion backend co Node.js y Express.js  Desarrollo frontend con React.js y tecnologías web.  Creación y consumo de APIs RESTful.  **3. Base de Datos y Persistencia.**  Diseño de modelos de datos relacionales.  Administración de bases de datos PostgreSQL.  Optimización de consultas y desempeño.  **4. Seguridad Informatica.**  implementación de sistemas de autenticación y autorización.  Gestión de roles y permisos de usuario.  Protección de datos y controles de acceso.  **5. Control de Calidad.**  Aplicación de pruebas de integración y validación  Revisión de código y estándares de desarrollo.  Documentación técnica y de usuario.  **6. Gestión de Configuracion**  Control de versiones con Git/GitHub.  Gestión de dependencias y entornos.  Deployment y configuración de servicios. | **Actividades Realizadas para el Desarrollo del Proyecto APT:**  **1. Análisis y planificación del Proyecto.**  Definición de requisitos y alcance del sistema.  Elaboración de backlog priorizado con historias de usuario.  Planificacion detallada del Sprint 1.  **2. Diseño de Arquitectura y Base de Datos.**  Diseño del modelo de datos relacional.  Configuración de base de datos PostgreSQL.  Definición de esquema de APIs RESTful.  **3. Desarrollo Backend.**  Implementación de sistema de autenticación JWT.  Desarrollo de APIs para gestión de usuarios.  Creacion de endpoints para operaciones CRUD de tickets.  Implementación de lógica de negocio y validaciones.  **4. Desarrollo Frontend**  Diseño de interfaz de usuario con React.js.  Implementación de componentes para gestión de tickets.  Desarrollo de formularios y validaciones del lado del cliente.  Creación de vistas responsivas y experiencia de usuario.  **5. Integración y Pruebas.**  Integración frontend-backend mediante APIs.  Pruebas de funcionalidad end-to-end.  Validación de flujos completos de usuario.  Pruebas de usabilidad y experiencia de usuario.  **6. Gestión y Documentación.**  Elaboración de documentación técnica y de usuario.  Seguimiento diario mediante Daily meetings.  Gestión de impedimentos y riesgos.  Preparación de entregables y presentaciones.  **Control de Calidad.**  Revisiones de código entre pares.  Pruebas de integración continua.  Validación de criterios de aceptación.  Aseguramiento de estándares de desarrollo. | **Recursos Utilizados para el Desarrollo del Proyecto:**  **1. Recursos Humanos:** Diego Inostroza: Scrum Master / Product Owner (15 horas).  Benjamín Barrera: Backend Developer (32 horas).  Juan Santander: Frontend Developer (19 horas).  **2. Recursos Tecnológicos.**  **Frontend:** React.js, HTML5, CSS3, JavaScript ES6+.  **Backend:** Node.js, Express.js, JWT para autenticacion.  **Base de Datos:** PostgreSQL 12+.  **Control de Versiones:** Git y GitHub.  **Entorno Desarrollo:** Visual Studio Code, Postman.  **3. Recursos de Infraestructura.**  Servidor de base de datos PostgreSQL.  Servidor de desarrollo para backend (Node.js).  Servidor web para frontend (React build).  Ambiente de testing y validación.  **4. Recursos de Gestión y Comunicación.**  Google Workspace (Docs, Sheets, Drive)  WhatsApp para comunicación inmediata.  Plantillas de documentación Scrum.  Herramientas de diagramación (Lucidchart).  **5. Recursos de documentación.**  Plantilla para documentos de gestión Scrum.  Estándares de documentación técnica.  Formatos para reportes de avance.  Repositorio de documentación en Google Drive.  **6. Recursos de Calidad.**  Herramientas de testing (Jest, Testing Library).  Estándares de coding y revisión de código.  Checklist de calidad para entregables.  Métricas de seguimiento de avance. | **Duración de Actividades del Sprint 1:**  **1. Análisis y Planificación del Proyecto.**  **Duración:** 3 días (25-27 ago 2025).  **Horas totales:** 8 horas.  **2. Diseño de Arquitectura y Base de Datos.**  **Duración:** 2 días (26-27 Ago 2025).  **Horas totales:** 12 horas.  **3. Desarrollo Backend.**  **Duración:** 5 días (27-31 Ago 2025).  **Horas totales:** 25 horas.  **4. Desarrollo Frontend.**  **Duración:** 5 días (27-31 Ago 2025).  **Horas totales:** 21 horas.  **5. Integración y Pruebas.**  **Duración:** 3 días (01-03 Sept 2025).  **Horas totales:** 21 horas.  **6. Gestión y Documentación.**  **Duración:** 10 días (25 Ago – 05 Sep 2025).  **Horas totales:** 9 horas.  **Resumen total:**  **Duración total del Sprint:** 10 días hábiles.  **Horas totales trabajadas:** 99 horas.  **Horas planificadas vs reales:** 66h vs 99h (+50%).  **Distribución por área:**  Desarrollo técnico: 58 horas (59%).  Gestión y documentación: 24 horas (24%).  Calidad y pruebas: 17 horas (17%). | **Asignación de Responsabilidades por Actividad:**  **1.Análisis y Planificación del Proyecto.**  **Responsable principal:** Diego Inostroza.  **Responsable apoyo:** Benjamín Barrera, Juan Santander.  **2. Diseño de Arquitectura y Base de Datos.**  **Responsable principal:** Benjamín Barrera  **Responsable apoyo:** Diego Inostroza.  **3. Desarrollo Backend.**  **Responsable principal:** Benjamín Barrera.  **Responsable apoyo:** Diego Inostroza (revisión).  **4. Desarrollo Frontend.**  **Responsable principal:** Juan Santander.  **Responsable apoyo:** Benjamín Barrera (APIs).  **5. Integración y Pruebas.**  **Responsable principal:** Benjamín Barrera y Juan Santander.  **Responsable apoyo:** Diego Inostroza (coordinación).  **6. Gestión y Documentación.**  **Responsable principal:** Diego Inostroza.  **Responsable apoyo:** Benjamín Barrera, Juan Santander.  **7. Control de Calidad**  **Responsable principal:** Equipo completo.  **Responsable principal:** Equipo completo.  **Revisión código backend:** Benjamín Barrera – Diego Inostroza.  **Revisión código frontend:** Juan Santander – Benjamín Barrera.  **Validación final:** Diego Inostroza.  **Distribución Horaria por responsable: Diego Inostroza:** 34 horas (34%).  **Benjamín Barrera:** 42 horas (42%).  **Juan Santander:** 23 horas (23%).  **Especialización por Rol:**  **Diego Inostroza:** Gestión, documentación, coordinación, revisión final.  **Benjamín Barrera:** Backend, base de datos, APIs, integración.  **Juan Santander:** Frontend, UI/UX, interfaz de usuario. | **Observaciones por Actividad:**  **1. Análisis y Planificación del Proyecto:**  **Facilitadores:** Experiencia previa en metodologías ágiles, claridad en los requisitos.  **Dificultades:** Complejidad en estimaciones iniciales por falta de historial de proyectos similares.  **2. Diseño de Arquitectura y Base de Datos.**  **Facilitadores:** Conocimiento sólido en PostgreSQL y diseño de modelos relacionales.  **Dificultades:** Bloqueo inicial por acceso a servidor de base de datos, resuelto en 4 horas.  **3. Desarrollo Backend.**  **Facilitadores:** Experiencia en Node.js/Express, documentación clara de APIs.  **Dificultades:** Complejidad mayor a la esperada en sistema de autenticación por roles.  **4. Desarrollo Frontend.**  **Facilitadores:** Habilidades en React.js, diseño de interfaces intuitivas.  **Dificultades:** Dependencia de end points backend para avanzar en integración.  **5. Integración y Pruebas.**  **Facilitadores:** Comunicación efectiva entre frontend y backend.  **Dificultades:** Issues de compatibilidad en actualización en tiempo real de estados.  **6. Gestión y Documentación.**  **Facilitadores:** Organización y disciplina en seguimiento diario.  **Dificultades:** Carga documental mayor a la anticipada inicialmente.  **7. Control de Calidad.**  **Facilitadores:** Cultura de revisión entre pares y estándares definidos.  **Dificultades:** Tiempo limitado para pruebas exhaustivas de todos los escenarios.  **Observaciones Generales:**  **Factores de éxito:** Comunicación diaria, roles definidos, compromiso del equipo.  **Lecciones aprendidas:** Necesidad de buffer en estimaciones, importancia de definir contratos técnicos tempranos.  **Riesgos mitigados:** Bloqueos técnicos resueltos mediante escalamiento inmediato y trabajo colaborativo. | **Estado de Avance por Actividad:**  **1. Análisis y Planificación del Proyecto.**  **Estado:** COMPLETADO.  **Porcentaje:** 100%.  **Fecha término:** 27 Ago 2025.  **2. Diseño de Arquitectura y Base de Datos.**  **Estado:** COMPLETADO.  **Porcentaje:** 100%.  **Fecha término:** 27 Ago 2025.  **3. Desarrollo Backend.**  **Estado:** COMPLETADO.  **Porcentaje:** 100%  **Fecha término:** 31 Ago 2025.  **4. Desarrollo Frontend.**  **Estado:** COMPLETADO.  **Porcentaje:** 100%  **Fecha término:** 31 Ago 2025  **5. Integración y Pruebas.**  **Estado:** COMPLETADO.  **Porcentaje:** 100%.  **Fecha término:** 03 Sep 2025.  **6. Gestión y Documentación.**  **Estado:** COMPLETADO.  **Porcentaje:** 100%  **Fecha término:** 05 Sep 2025.  **7. Control de Calidad.**  **Estado:** COMPLETADO.  **Porcentaje:** 100%.  **Fecha término:** 05 Sep 2025.  **Resumen General del Estado:**  **Totales actividades:** 7.  **Completadas:** 7 (100%).  **En curso:** 0.  **Retrasadas:** 0.  **No iniciadas:** 0.  **Métricas de Avance:**  **Horas planificadas:** 66 horas.  **Horas ejecutadas:** 99 horas.  **Tasa de cumplimiento:** 100% en alcance, 150% en esfuerzo.  **Sprint 1:** ENTREGADO EXITOSAMENTE. | **Ajustes Realizados al Plan de Trabajo:**  **1. Análisis y Planificación del Proyecto.**  **Ajuste:** Incremento de 2 horas en estimación inicial.  **Justificación:** Necesidad de mayor análisis de dependencias técnicas.  **2. Diseño de Arquitectura y Base de Datos.**  **Ajuste:** Replanificación por bloqueo de acceso a bd (+1 día).  **Justificación:** Bloqueo técnico requirió gestión adicional de recursos.  **3. Desarrollo Backend.**  **Ajuste:** Aumento de 5 horas en desarrollo de autenticación.  **Justificación:** Complejidad mayor en sistema de roles y permisos.  **4. Desarrollo Frontend.**  **Ajuste:** Extensión de 3 horas en integración con APIs.  **Justificación:** Dependencia de endpoints backend y ajustes en contratos API.  **5. Integración y Pruebas.**  **Ajuste:** Adición de 4 horas para resolver issues de compatibilidad.  **Justificación:** Problemas de actualización en tiempo real requirieron solución con polling.  **6. Gestión y Documentación.**  **Ajuste:** Incremento de 3 horas en documentación más exhaustiva del proceso.  **7. Control de Calidad.**  **Ajuste:** Adición de 2 horas para pruebas cross-browser.  **Justificación:** Issues de compatibilidad con Safari requirieron ajustes específicos.  **Ajustes Metodológicos Implementados:** Buffer del 20%: incorporado en estimaciones futuras.  **Reuniones técnicas:** 2 veces por semana añadidas al proceso.  **Contratos API formales:** Establecidos para futuros sprints.  **Registro de impedimentos:** Formalizado como proceso estándar.  **Justificación de Ajustes:**  Los ajustes fueron necesarios debido a:  Complejidades técnicas no anticipadas en planning inicial.  Dependencias entre frontend-backend más críticas de lo esperado.  Requerimientos de calidad y documentación adicionales.  Bloqueos de recursos externos no controlables por el equipo. |

|  |
| --- |
| **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:  **Factores Facilitadores:**  **1. Comunicación Efectiva**   * Reuniones diarias de 15 minutos que permitieron identificar y resolver bloqueos rápidamente * Canal de WhatsApp para comunicación inmediata y coordinación en tiempo real * Transparencia en el reporte de avances y dificultades por parte de todo el equipo   **2. Roles y Especializaciones Definidas**   * Clara distribución de responsabilidades según competencias técnicas * Benjamin (backend/BD), Juan (frontend/UI), Diego (gestión/documentación) * Autonomía para tomar decisiones técnicas dentro de cada área de especialización   **3. Metodología Ágil Bien Implementada**   * Ceremonias Scrum aplicadas consistentemente * Seguimiento diario mediante Excel de métricas * Flexibilidad para ajustar el plan basado en lecciones aprendidas   **4. Documentación Organizada**   * Sistema centralizado en Google Drive para todos los documentos * Plantillas estandarizadas que agilizaron la elaboración de reportes * Historial completo de decisiones y cambios   **Factores que Dificultaron el Desarrollo:**  **1. Estimaciones Optimistas**   * Subestimación de la complejidad técnica en desarrollo de autenticación * No considerar suficiente tiempo para integración frontend-backend * Falta de buffer para imprevistos técnicos   **Acciones tomadas:** Implementación de buffer del 20% para futuras estimaciones y uso de planning poker con todo el equipo.  **2. Bloqueos Técnicos No Anticipados**   * Problemas de acceso a servidor de base de datos (4 horas de bloqueo) * Issues de compatibilidad cross-browser con Safari * Dependencias técnicas no identificadas en planning inicial   **Acciones tomadas:** Establecimiento de registro formal de impedimentos y protocolo de escalamiento inmediato.  **3. Dependencias entre Componentes**   * Frontend dependiente de endpoints backend para avanzar * Retrasos en integración por diferencias en contratos API * necesidad de sincronización constante entre equipos técnicos   **Acciones tomadas:** Implementación de reuniones técnicas 2 veces por semana y definición de contratos API formales.  **4. Carga Documental**   * Tiempo requerido para documentación mayor al anticipado * Necesidad de mantener actualizados múltiples documentos simultáneamente * Balance difícil entre desarrollo y documentación   **Acciones tomadas:** Optimización de plantillas y distribución más equilibrada de tareas documentales. |

|  |
| --- |
| Actividades ajustadas o eliminadas:  **Actividades Ajustadas:**  **1. Desarrollo del Sistema de Autenticación**   * **Ajuste realizado:** Incremento de 5 horas en la estimación original * **Justificación:** La implementación de control de acceso por roles (usuario/técnico/supervisor) resultó más compleja de lo anticipado, requiriendo desarrollo adicional de middlewares de autorización   **2. Integración Frontend-Backend**   * **Ajuste realizado:** Adición de 4 horas para pruebas de integración * **Justificación:** Surgieron issues de compatibilidad en la actualización en tiempo real de estados, requiriendo implementación de sistema de polling cada 5 segundos   **3. Pruebas Cross-Browser**   * **Ajuste realizado:** Incorporación de 2 horas específicas para compatibilidad * **Justificación:** Se identificaron problemas de visualización en Safari que requirieron implementación de polyfills y ajustes CSS específicos   **4. Documentación Técnica**   * **Ajuste realizado:** Extensión de 3 horas en la documentación de APIs * **Justificación:** Se necesitó documentar exhaustivamente los endpoints para facilitar la integración y mantenimiento futuro   **Actividades No Eliminadas:**  No se eliminó ninguna actividad del plan original debido a que todas eran esenciales para el MVP. Sin embargo, se priorizó el cumplimiento del 100% de las funcionalidades core, manteniendo algunas características secundarias para sprints futuros.  **Justificación General de los Ajustes:**  Los ajustes fueron necesarios para mantener la calidad del producto final mientras se enfrentaban desafíos técnicos no anticipados. La decisión de ajustar en lugar de eliminar actividades se basó en:   * Compromiso con la entrega de un producto funcional y de calidad * Identificación temprana de desviaciones mediante seguimiento diario * Capacidad del equipo para absorber carga adicional cuando fue necesario * Enfoque en el valor entregable sobre el cumplimiento estricto de estimaciones iniciales   **Resultado:** Todos los objetivos funcionales se cumplieron al 100%, demostrando que los ajustes fueron apropiados y efectivos. |

|  |
| --- |
| Actividades que no has iniciado o están retrasadas:  **No existen actividades retrasadas o no iniciadas en el Sprint 1.** Todas las 7 actividades planificadas fueron completadas dentro del período establecido (25 Ago - 05 Sep 2025), cumpliendo el 100% de los objetivos del sprint.  **Estrategias que permitieron este cumplimiento:**  **1. Gestión Proactiva de Tiempos**   * Seguimiento diario riguroso mediante Daily Meetings * Identificación temprana de posibles retrasos * Redistribución inmediata de recursos cuando fue necesario   **2. Comunicación Efectiva**   * Reporte transparente de avances y dificultades * Coordinación constante entre los miembros del equipo * Resolución colaborativa de problemas   **3. Flexibilidad en la Ejecución**   * Ajuste de horarios y carga de trabajo cuando se identificaron cuellos de botella * Capacidad del equipo para absorber carga adicional * Reorganización de prioridades en tiempo real   **4. Compromiso del Equipo**   * Dedicación de horas adicionales para cumplir con los plazos * Enfoque colectivo en los objetivos comunes * Cultura de responsabilidad y ownership   **El cumplimiento del 100% de las actividades demuestra la efectividad de la metodología Scrum aplicada y la capacidad del equipo para ejecutar el plan de trabajo según lo establecido.** |

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-1)