}

1. **PARTE I**

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

| Nombre estudiante | **Alejandro Villa Villavicencio, Marcelo Jesus Madriaga Peña, Matías General Muñoz** |
| --- | --- |
| Rut | **18.248.569-1, 17.670.931-6, 19.783.934-1** |
| Carrera | **Ingeniería en informática** |
| Sede | **San Bernardo** |

| **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 | *Sistema Unidad Territorial* |
| --- | --- |
| Área (s) de desempeño(s) | *Análisis y evaluación de soluciones informáticas*  *Desarrollo de Software* |
| Competencias | *Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento asegurando el logro de los objetivos.*  *Construir modelos de datos para soportar los requerimientos de la organización de acuerdo a un diseño definido y escalable en el tiempo.*  *Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria.* |

| **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 siguiente proyecto busca abordar la necesidad de digitalizar y mejorar la gestión de actividades y recursos en comunidades locales (Junta de vecinos), Este problema es relevante porque muchas comunidades aún dependen de procesos manuales o herramientas no centralizadas para coordinar eventos, gestionar presupuestos y comunicarse eficientemente.  Este proyecto es simulado, el cual aborda a una comunidad ficticia de la comuna de San Bernardo.  El aporte a la comunidad se ve afectada directamente ya que proporcionará una mejora en la calidad de esta, al poder entregar la documentación requerida mas eficientemente, las solicitudes de espacios comunes de manera más ágil además de la postulación de proyectos, sin duda mejorará la calidad de vida de las personas de la comunidad ya que estas comúnmente suelen tener recursos limitados y una estructura organizativa basada en voluntarios lo cual normalmente es difícil de poder acceder a dichas personas, para realizar dichas solicitudes las cuales serán digitalizadas. |
| --- | --- |
| *Descripción del Proyecto APT* | Objetivo:  El objetivo del proyecto es desarrollar una aplicación web y móvil que permita a las juntas de vecinos gestionar sus actividades, recursos, y comunicaciones de manera más eficiente y accesible para toda la comunidad.  Descripción del Proyecto:  El proyecto consistirá en el desarrollo de un sistema utilizando NodeJS como backend y Firebase para la seguridad de inicio de sesión y el almacenamiento de datos. El frontend web será desarrollado con Ionic y Angular. El sistema permitirá la gestión de eventos, la administración de presupuestos, la comunicación interna, y la difusión de noticias relevantes para la comunidad. La app móvil también será desarrollada en Ionic y Angular, proporcionando una experiencia fluida y multiplataforma.  Enfoque:  Para abordar la problemática, se desarrollará el sistema en 6 sprints utilizando la metodología Scrum. Los sprints incluirán la toma de requisitos, el diseño del sistema, la implementación del backend con NodeJS, la integración con Firebase para la autenticación y almacenamiento de datos, y el desarrollo del frontend para web y móvil con Ionic y Angular. Cada sprint será revisado y ajustado según el feedback de los usuarios finales y los stakeholders. |
| *Pertinencia del proyecto con el perfil de egreso* | Relación con el Perfil de Egreso:  Este proyecto está alineado con el perfil de egreso , ya que involucra competencias las cuales son claves como el análisis y diseño de sistemas de información, desarrollo de aplicaciones web y móviles, y la implementación de soluciones tecnológicas.  Competencias Necesarias:  Las competencias en áreas como análisis, diseño, y desarrollo de sistemas son fundamentales para este proyecto, ya que permiten identificar correctamente los requisitos del sistema, diseñar una arquitectura adecuada y desarrollar una solución que cumpla con las necesidades de los usuarios. Además, la gestión de proyectos y el trabajo en equipo son esenciales para asegurar la entrega del proyecto de manera exitosa. |
| *Relación con los intereses profesionales* | Este proyecto está enfocado al desarrollo de soluciones tecnológicas que integren tanto el diseño como la implementación de sistemas software, especialmente en áreas que promuevan la eficiencia y la mejora en la gestión de procesos. Está particularmente interesado en proyectos que impacten positivamente a la comunidad, aplicando metodologías ágiles como Scrum. Además, tiene un gran interés en automatizar procesos y crear plataformas que faciliten la toma de decisiones y la organización a nivel comunitario.  El proyecto se centra en la creación de un sistema para mejorar la gestión de una comunidad o junta vecinal en San Bernardo y está directamente alineado con los intereses profesionales. Este proyecto no sólo implica el diseño y desarrollo de software, sino que también tiene un fuerte componente de impacto social, al facilitar la organización y gestión de recursos a nivel comunitario.  Experiencia práctica en desarrollo de software.  Aplicación de Metodologías Ágiles  Impacto social y comunitario  Integración de Nuevas Tecnologías |
| *Factibilidad de desarrollo del Proyecto APT* | Duración del Semestre:  El proyecto es factible de realizar dentro del tiempo asignado, ya que está planificado para completarse en seis sprints. Este enfoque permite una mejor gestión del tiempo y los recursos, asegurando que cada fase del desarrollo (requisitos, diseño, implementación, pruebas, y ajustes) esté bien distribuida a lo largo del semestre.  Horas Asignadas a la Asignatura:  Las horas asignadas a la asignatura serán suficientes para completar las tareas de cada sprint, especialmente si se sigue un cronograma bien definido. Los seis sprints permiten un enfoque iterativo, asegurando que cada fase pueda ajustarse según el feedback de los stakeholders.  Materiales Requeridos:  Los materiales necesarios incluyen herramientas de desarrollo como un IDE adecuado para NodeJS, Angular e Ionic, acceso a Firebase para la autenticación y almacenamiento de datos, y dispositivos Android para probar la aplicación móvil. Todos estos recursos están disponibles y son accesibles.  Factores Externos Facilitadores:  El uso de frameworks y tecnologías modernas como NodeJS, Ionic, Angular, y Firebase facilita el desarrollo, permitiendo escalabilidad y eficiencia. En conjunto con la metodología Scrum, el proyecto puede ajustarse fácilmente en caso de que surjan problemas.  Factores Externos: Dificultades y Soluciones:  Posibles problemas incluyen compatibilidad entre diferentes versiones de Android y la disponibilidad de tiempo de los miembros del equipo. Estos problemas pueden mitigarse mediante pruebas tempranas en múltiples dispositivos y una buena comunicación dentro del equipo, asegurando que todos estén alineados con los plazos y objetivos de cada sprint. |

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 | Desarrollar una solución tecnológica integral que permita a las juntas de vecinos gestionar de manera eficiente sus actividades, recursos y la comunicación a través de una aplicación web y móvil, mejorando así la organización y participación dentro de la comunidad. |
| --- | --- |
| Objetivos específicos | * Diseñar la arquitectura del sistema, incluyendo el backend, frontend, y la aplicación móvil, asegurando la escalabilidad y la seguridad de la información mediante la integración de NodeJS y Firebase para la autenticación y el almacenamiento de datos. * Implementar el backend utilizando NodeJS y Firebase para gestionar las conexiones, procesos del servidor, y el almacenamiento, garantizando un rendimiento óptimo y una gestión segura de los datos. * Desarrollar la interfaz web y móvil utilizando Ionic y Angular, asegurando un diseño responsive y accesible que facilite la interacción de los usuarios con el sistema en diferentes dispositivos y plataformas. * Crear una aplicación móvil compatible con dispositivos Android 10 en adelante, que permita a los usuarios gestionar actividades, recibir notificaciones en tiempo real, y realizar trámites de manera eficiente. * Realizar pruebas de usabilidad y funcionalidad en diferentes versiones de Android y navegadores web para asegurar la compatibilidad y el buen desempeño del sistema en todos los entornos. * Documentar cada fase del desarrollo del proyecto, incluyendo la toma de requisitos, diseño, implementación, pruebas, y entrega final, para garantizar la trazabilidad y transparencia en el proceso 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 |
| --- |
| La metodología a utilizar será Scrum, dividiendo el proyecto en varios sprints que cubrirán desde la toma de requisitos hasta la implementación final. Se realizará una revisión al final de cada sprint para ajustar el desarrollo según la retroalimentación de los stakeholders. El trabajo será colaborativo, con roles claramente definidos para asegurar la eficiencia del equipo. |

| **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**  **(avance o final)** | **Nombre de la evidencia** | **Descripción** | **Justificación** |
| --- | --- | --- | --- |
| **Avance** | **Acta de constitución** | *Formaliza el inicio legal del proyecto, esto nos proporciona una vista general del proyecto, tal como el alcance, los límites del proyecto, las fechas y los costos del proyecto. Tiene como* | **Es el documento de partida del proyecto en el cual se establecen las bases legales, costos, involucrados, alcances entre otros.** |
| **Avance** | **Kick Off** | *Reunión inicial que marca el comienzo formal del proyecto, el objetivo principal es alinear el equipo de trabajo con los objetivos, alcance, roles, responsabilidades y planificación general* | **Se implementa el kick off para alinear al equipo y clarificar las expectativas.** |
| **Avance** | **Documento de requerimientos** | *Detalla las funcionalidades, servicios y restricciones operativas de un sistema* | **Documenta que emplea los requerimientos del sistema, el que se realizara y como se deberá realizar.** |
| **Avance** | **Plan de gestión de riesgos** | *Describe cómo se identificarán, analizarán, monitorearán y controlarán los riesgos durante la vida del proyecto.* | **Este plan nos indica los riesgos asociados al proyecto en cada fase o etapa del desarrollo de este.** |
| **Avance** | **Diagrama de modelo de datos no relacional** | *Nos muestra cómo se va a organizar y relacionar entre sí los datos sin utilizar una estructura de tabla tradicional* | **Diagrama que muestra la construcción de la base de datos no relacional para mejorar su entendimiento.** |
| **Avance** | **Diagrama de clases** | *Es la representación visual de las estructuras y relaciones entre clases en la programación orientada a objetos. Muestra las clases, atributos, métodos y las interacciones entre ellas. Fundamental para definir la arquitectura del sistema y facilitar el entendimiento del código y su mantenimiento* | **Se utiliza para representar de manera visual las clases y sus relaciones dentro del sistema.** |
| **Final** | **Historias de usuario** | *Sirven para comunicar como una función específica entregará valor al usuario* | **Su utilización es para comunicar y capturar los requisitos del usuario enfocándose en la funcionalidad.** |
| **Final** | **Codigo fuente página web** | *Elementos, estructura y funcionalidad que compone el sistema informático solicitado.* | **El código fuente nos muestra como está creado y cómo se comporta el sistema solicitado, esto nos ayuda para futuros cambios y mejoras en el.** |
| **Final** | **Código fuente aplicación móvil** | *Elementos, estructura y funcionalidad que compone el sistema informático solicitado.* | **El código fuente de la aplicación móvil muestra la construcción del sistema solicitado, esto también ayuda a la transparencia de la información obtenida y entregada en un dispositivo móvil** |
| **Avance** | **Plan de Sprints** | *Detalla la distribución del trabajo en intervalos cortos y fijos, conocidos como Sprints, que suelen durar entre una a cuatro semanas.* | **Se utiliza para organizar el trabajo en ciclos cortos, permitiendo dividir el proyecto en partes manejables, asegurar entregas continuas de valor, y facilitar la adaptación a cambios durante el desarrollo.** |
| **Avance** | **Prototipos interactivos** | *Muestra una aproximación de lo que sería el sistema final* | **Es necesario la visualización temprana e interactiva para que usuario final y el cliente tengan una mejor participación en el proyecto e ir mejorando dentro del desarrollo.** |
| **Avance** | **Encuestas de satisfacción de usuarios** | *Sirve para medir cómo perciben los usuarios o clientes el producto en términos de usabilidad, funcionalidad y cumplimiento de expectativas.* | **Ayuda a identificar áreas de mejora y asegura que el sistema realmente cumple con las necesidades del usuario, lo que es crucial para el éxito y la aceptación del producto** |
| **Final** | **Bitácora de errores** | *Registro detallado donde se documentan todos los problemas, fallos o errores que se encuentran durante el desarrollo, prueba o uso de un sistema informático.* | **Permite rastrear y analizar problemas recurrentes, facilitar la resolución de errores, y mejorar la calidad del sistema al documentar las soluciones aplicadas.** |

| **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** |
| *Gestión de proyectos.* | *Guía de definición del proyecto* | *Detalla la estructura y los requerimientos del proyecto.* | *Equipo de trabajo, Stakeholders.* | *Una semana* | *Todos* | *Mala coordinación del equipo. Mala comunicación.* |
| *Gestión de proyectos. Comunicación eficaz.* | *Acta de constitución* | *Formaliza el inicio del proyecto, proporcionando una visión general y definiendo la autoridad del jefe de proyecto.* | *Software de gestión de proyectos, herramientas de comunicación* | *3 días a una semana* | *Alejandro Villa* | *Cambios en los requerimientos. Ser poco claros en las revisiones. Limitación de recursos financieros.* |
| *Análisis de requerimientos.* | *ERS (Especificación de Requisitos de Software)* | *Detalla las funcionalidades, servicios y restricciones operativas del sistema.* | *Herramientas de comunicación.* | *Una semana* | *Alejandro Villa* | *Ambigüedad en los requerimientos. Cambios en los requerimientos. Mala comunicación. Restricciones de presupuesto y tiempo.* |
| *Diseño de base de datos no relacionales.* | *Diagrama de datos no relacionales* | *Visualiza la organización de los datos para optimizar rendimiento y escalabilidad en Firebase.* | *Firebase, equipo de trabajo.* | *De una a dos semanas* | *Alejandro Villa* | *Falta de experiencia. Cambios en los requerimientos.* |
| *Gestión ágil de proyectos.* | *Plan de Sprint (Sprint Planning)* | *Define las tareas a completar en cada sprint para mantener el ritmo y entregar resultados incrementales.* | *Herramientas de gestión de proyectos.* | *Variable según cada sprint* | *Marcelo Madriaga* | *Mala estimación de tareas. Dificultad en la priorización de tareas. No adherirse a los tiempos definidos.* |
| *Evaluación de riesgos.* | *Plan de Gestión de Riesgos* | *Documento que identifica, analiza y gestiona los riesgos a lo largo del proyecto.* | *Herramientas de gestión de riesgos (Excel).* | *Una semana* | *Marcelo Madriaga* | *Mala identificación de riesgos. Mitigación poco efectiva.* |
| *Comunicación efectiva.* | *Kickoff* | *Reunión inicial que marca el comienzo formal del proyecto.* | *Equipo de trabajo, Stakeholders.* | *Un día* | *Matías General* | *Ausencia de algún stakeholder o miembro clave del equipo.* |
| *Gestión ágil de proyectos.* | *Product Backlog* | *Lista priorizada de todos los requerimientos y tareas del proyecto que aún no han sido asignadas a un sprint.* | *Herramientas de gestión de proyectos.* | *Continuo* | *Alejandro Villa* | *Sobrecarga de ítems*  *Definiciones pobres de historias de usuario*  *Estimaciones inexactas*  *Cambios frecuentes*  *No mantener actualizado el product backlog* |
| *Gestión ágil de proyectos.* | *Sprint Backlog* | *Lista priorizada de tareas seleccionadas para el sprint en curso, con estimaciones de tiempo y esfuerzo.* | *Herramientas de gestión de proyectos.* | *Continuo* | *Marcelo Madriaga* | *Sobrecarga de tareas.*  *No priorizar correctamente las tareas.*  *Dependencias entre tareas no identificadas.*  *Cambios durante el sprint.*  *Falta de claridad en las tareas.*  *Poca capacidad del equipo.* |
| *Incremento de Producto.* | *Incremento de Producto* | *Es el resultado del trabajo completado en cada sprint, que debe estar listo para ser utilizado o entregado.* | *Herramientas de desarrollo.* | *Continuo* | *Marcelo Madriaga* | *Malas regresiones pueden introducir errores en partes del software.* |
| *Documentación técnica.* | *Diagrama de Arquitectura del Sistema (Diagrama de Despliegue)* | *Diagrama que representa la infraestructura de hardware y cómo interactúan los componentes del sistema.* | *Software para diagramas.*  *(Lucidchart, Draw.io).* | *Dos días* | *Matías General* | *Errores en la implementación de estándares, herramientas inadecuadas. Cambios en los requisitos. Comunicación ineficiente.* |
| *Documentación técnica.* | *Diagrama de Componentes* | *Diagrama que muestra la organización y relaciones entre los componentes del sistema.* | *Software para diagramas.* | *Dos días* | *Alejandro Villa* | *Usar herramientas que no conoce su uso.*  *Mala identificación de componentes.*  *Dependencias no documentadas o mal entendidas.*  *No tener conocimiento del desarrollo* |
| *Documentación técnica.* | *Diagrama de Clases* | *Diagrama que organiza las clases de la aplicación (si se usa POO), mostrando atributos, métodos y relaciones.* | *Software para diagramas.* | *Dos días* | *Alejandro Villa* | *Usar herramientas que no conoce su uso.*  *Mala identificación de componentes.*  *Dependencias no documentadas o mal entendidas.*  *No tener conocimiento del desarrollo* |
| *Documentación técnica.* | *Diagrama de Secuencia* | *Diagrama que muestra cómo los objetos interactúan en una secuencia de tiempo.* | *Software para diagramas.* | *Dos días* | *Matias General* | *Usar herramientas que no conoce su uso.*  *Mala identificación de componentes.*  *Dependencias no documentadas o mal entendidas.*  *No tener conocimiento del desarrollo* |
| *Documentación técnica.* | *Diagrama de Flujo de Datos (DFD)* | *Representa el flujo de información dentro del sistema, incluyendo entradas, salidas, y procesos.* | *Software para diagramas de flujo.* | *Dos días* | *Matias General* | *Usar herramientas que no conoce su uso.*  *Mala identificación de componentes.*  *Dependencias no documentadas o mal entendidas.*  *No tener conocimiento del desarrollo* |
| *Documentación técnica.* | *Diagrama de Casos de Uso* | *Diagrama que muestra las interacciones entre los actores (usuarios) y las funcionalidades del sistema.* | *Software para diagramas de casos de uso.* | *Dos días* | *Todos* | *Usar herramientas que no conoce su uso.*  *Mala identificación de componentes.*  *Dependencias no documentadas o mal entendidas.*  *No tener conocimiento del desarrollo* |
| *Documentación técnica.* | *Diagrama de Procesos* | *Muestra los procesos internos del sistema, incluyendo validaciones y cálculos que se realizan en segundo plano.* | *Software de diagramas.* | *Dos días* | *Todos* | *Usar herramientas que no conoce su uso.*  *Mala identificación de componentes.*  *Dependencias no documentadas o mal entendidas.*  *No tener conocimiento del desarrollo* |
| *Scrum.* | *Evidencia de Ceremonias Scrum* | *Recolección de las actas o minutas de reuniones de planificación de sprint, revisiones, retrospectivas, y reuniones diarias.* | *Herramientas de gestión de proyectos.* | *Continuo* |  | *Falta de participación o compromiso del equipo de trabajo..*  *Mala gestión del tiempo.*  *Poca preparación de los integrantes del equipo.*  *Mala comunicación.* |
| *Scrum.* | *Tablero Scrum o Kanban* | *Tablero que muestra el progreso del equipo durante cada sprint, gestionando las tareas pendientes, en progreso y completadas.* | *Herramientas de Kanban (Trello, Jira, etc.).* | *Continuo* |  | *No mantener el tablero actualizado.*  *Sobrecarga de información.*  *Poco compromiso del equipo de trabajo*  *Confiar demasiado en el tablero como única fuente de información.*  *Uso incorrecto del tablero.* |

| **8. Cronograma Sprint.** |
| --- |
| 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. |

Se detalla la planificación de los sprint para el proyecto, el cual se organiza en ciclos cortos, para garantizar la entrega continua de valor y adaptación constante a los cambios.

Cada sprint tiene una duración específica y se enfoca en diferentes aspectos del desarrollo, desde la definición y planificación inicial hasta la entrega final y evaluación del proyecto. A través de reuniones diarias y revisiones regulares, el equipo asegura la calidad y el progreso continuo del proyecto, manteniendo una alineación constante con los objetivos del cliente y los requisitos del sistema.

### **Sprint 1**

**Duración y Fechas**: 22-08-24 a 05-09-24 (10 días hábiles)

* **Actividades**:
  + Definición del Product Backlog
  + Identificación y Priorización de Historias de Usuario
  + Estimación de Esfuerzos
  + Planificación del Sprint (Sprint Planning)
  + Asignación de Tareas para el Sprint
  + Daily Scrum Meetings

### **Sprint 2**

**Duración y Fechas**: 06-09-24 a 10-10-24 (23 días hábiles)

* **Actividades**:
  + Desarrollo Iterativo del Producto
  + Implementación de Funcionalidades
  + Revisión de los Criterios de Aceptación de Historias de Usuario
  + Revisión de Código y Colaboración en Pares
  + Daily Scrum Meetings
  + Identificación y Evaluación de Riesgos (continuo)

### 

### **Sprint 3**

**Duración y Fechas**: 11-10-24 a 17-10-24 (5 días hábiles)

* **Actividades**:
  + Ejecución de Pruebas Unitarias e Integración
  + Revisión de Pruebas de Aceptación (User Acceptance Testing - UAT)
  + Ajustes basados en el Feedback de las Pruebas
  + Actualización del Backlog con Mejoras y Correcciones
  + Daily Scrum Meetings

### **Sprint 4**

**Duración y Fechas**: 18-10-24 a 31-10-24 (10 días hábiles)

* **Actividades**:
  + Despliegue del Incremento en un Entorno de Pruebas o Producción
  + Capacitación a Usuarios Finales
  + Documentación del Uso del Sistema
  + Revisión de la Retroalimentación del Usuario (Feedback Loop)
  + Daily Scrum Meetings

### **Sprint 5**

**Duración y Fechas**: 01-11-24 a 14-11-24 (10 días hábiles)

* **Actividades**:
  + Monitoreo del Rendimiento y Uso del Sistema
  + Identificación de Áreas de Mejora
  + Incorporación de Mejoras en el Product Backlog
  + Daily Scrum Meetings

### **Sprint 6 Final**

**Duración y Fechas**: 15-11-24 a 02-12-24 (12 días hábiles)

* **Actividades**:
  + Revisión Final del Proyecto y Confirmación de Entrega (Sprint Review)
  + Documentación Final del Proyecto
  + Evaluación Retrospectiva del Proyecto (Sprint Retrospective)
  + Lecciones Aprendidas y Cierre Formal del Proyecto
  + Daily Scrum Meetings

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)