**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 | *Se han llevado a cabo todas las actividades de forma completa según el cronograma hasta las tareas de pruebas, todo lo anterior a estas tareas están totalmente completas, y revisado por el profesor guía.* |
| Objetivos | *Mejorar la gestión de la unidad territorial de una junta de vecinos a través del desarrollo e implementación de una plataforma web.* |
| Metodología | *El proyecto se abordará utilizando la metodología tradicional en cascada, la cual se caracteriza por un desarrollo secuencial y estructurado, avanzando por etapas bien definidas y con entregables en cada una de ellas. Esta metodología resulta adecuada para el contexto de una Junta de Vecinos, ya que permite tener claridad sobre las fases, objetivos y plazos, asegurando un producto final alineado a los requerimientos iniciales.* |
| Evidencias de avance | Fase de Avance: Esta etapa se centra en la documentación rigurosa que sustenta el diseño y la estrategia de calidad. Las evidencias clave son los Documentos de Planificación y Diseño (arquitectura, diagramas de base de datos, mockups de interfaz y casos de uso), cuya justificación principal es validar y asegurar que el sistema esté correctamente diseñado antes de la implementación. Asimismo, se requiere la entrega de los Documentos de Control y Pruebas, que detallan el plan de validación y los escenarios de testing, garantizando que el sistema cumpla con todos los requerimientos funcionales y no funcionales desde las primeras etapas.  Fase Final (Cierre): El cierre del proyecto se demuestra a través de dos entregables fundamentales. La primera y más importante es el Sistema (Web y Base de Datos) operativo, que representa la evidencia tangible del logro de los objetivos del plan inicial. El segundo entregable es el Documento de Cierre, un informe final que resume los resultados, detalla las dificultades encontradas, expone las lecciones aprendidas y presenta las conclusiones generales del proyecto, ofreciendo una visión integral y documentada del proceso. |

|  |
| --- |
| **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 Proyecto APT** | | | | | | |
| Competencia / Unidad de Competencia | Nombre de Actividad / Tarea | Descripción Actividades / Tareas | Recursos | Duración | Responsable | Observaciones |
| Análisis de Requerimientos | Levantamiento de Requerimientos | Reunir y documentar necesidades del cliente sobre el sistema | Reuniones virtuales, documentos Word/Excel, entrevistas. | 1 semana | Analista de sistemas | Posible dificultad: falta de claridad en requerimientos del cliente. |
| Análisis de Requerimientos | Validación de Requerimientos | Confirmar con el cliente que los requerimientos documentados reflejan sus necesidades. | Documentación de requerimientos, sesiones de revisión. | 2 días | Analista de sistemas | Riesgo de cambios posteriores si no se valida correctamente. |
| Diseño del Sistema | Diseño Funcional | Crear diagramas de casos de uso, modelo entidad-relación y diagramas de flujo. | Herramientas UML (Draw.io, Lucidchart). | 1 semana | Diseñador de sistemas | Facilita la comprensión del sistema, riesgo de ajustes por cambios menores. |
| Diseño del Sistema | Diseño Técnico | Definir arquitectura, base de datos, tecnologías front-end y back-end. | Documentación técnica, herramientas de modelado. | 1 semana | Arquitecto de software | Dificultad: selección inadecuada de tecnologías que impacte escalabilidad. |
| Desarrollo | Programación Front-End | Implementar interfaz gráfica con HTML, CSS, JS y framework definido. | VS Code, librerías, GitHub. | 2 semanas | Desarrollador Front-End | Posibles retrasos por integración de diseño responsivo. |
| Desarrollo | Programación Back-End | Desarrollar lógica de negocio, API y conexión a la base de datos. | Lenguaje de programación ,servidor, Git. | 3 semanas | Desarrollador Back-End | Riesgo: errores en la integración con la base de datos. |
| Desarrollo | Integración de Módulos | Unir componentes front-end y back-end para un funcionamiento completo. | GitHub, Postman. | 1 semana | Equipo de desarrollo | Posible dificultad en compatibilidad de versiones o dependencias. |
| Pruebas | Pruebas Unitarias | Verificar funcionalidad de cada módulo desarrollado. | PyTest y selenium. | 1 semana | QA Tester | Puede requerir correcciones que retrasen la siguiente fase. |
| Pruebas | Pruebas de Integración y Aceptación | Validar el sistema completo con datos de prueba y con los usuarios. | Entorno de pruebas | 1 semana | QA |  |
| Implementación y cierre | Despliegue del Sistema | Instalar el sistema en el servidor productivo y configurar accesos. | Hosting, base de datos en la nube, credenciales. | 3 días | DevOps | Posible retraso por problemas en la infraestructura de hosting. |
| Implementación y cierre | Capacitación a Usuarios | Entrenar a los usuarios en el uso del sistema | Manual de usuario, presentaciones, sesiones de capacitación. | 3 días | Líder de proyecto | Posible resistencia de usuarios finales al cambio. |
| Mantenimiento | Corrección de Errores Iniciales | Atender incidencias detectadas tras el despliegue. | Sistema de tickets, equipo de soporte. | 5días(post-entrega) | Equipo de soporte | Riesgo: alta demanda de soporte inicial. |

|  |
| --- |
| **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 Facilitadores**  El desarrollo del proyecto fue altamente eficiente gracias a varios factores clave que permitieron el cumplimiento del 100% de las actividades de programación antes del plazo original:   1. **Adherencia Estricta al Modelo Cascada:** La planificación rígida y detallada, basada en la documentación previa (DAS y ERS), eliminó ambigüedades. Esto permitió una transición fluida y lineal desde el diseño hasta la codificación, evitando desviaciones de alcance. 2. **Decisiones Tecnológicas Robustas:** La elección del *stack* Django/Python demostró ser una base sólida, acelerando la implementación de la lógica de negocio, la gestión de la base de datos y la seguridad, lo que contribuyó a la velocidad general de la fase de desarrollo. 3. **Auto-Gestión y Enfoque Individual:** Al ser un proyecto individual, el alto nivel de autodisciplina y el control directo sobre todas las capas del desarrollo (Front-end, Back-end, DB) resultaron en una toma de decisiones inmediata y una ejecución sin demoras por dependencias de equipo.   **Factores que han dificultado el desarrollo**  El factor de dificultad más significativo estuvo asociado a la integración de nuevas tecnologías de terceros:   1. **Curva de Aprendizaje en Integraciones:**   La implementación de herramientas externas como **n8n** (para notificaciones automatizadas de WhatsApp y Email) y la librería para la **generación de documentos PDF** en Django requirieron un tiempo de investigación y autoaprendizaje considerable, lo cual puso a prueba los plazos internos de la Carta Gantt.   1. **Falta de Retroalimentación Inmediata:**   La ausencia de un par o revisor de código externo significó que la detección de errores de lógica dependió exclusiv amente de las pruebas unitarias realizadas por el propio desarrollador. |

|  |
| --- |
| Actividades ajustadas o eliminadas:  La metodología **Cascada**, combinada con la planificación detallada y la asignación de tiempo de amortiguación (buffers), fue la clave para mitigar las dificultades (especialmente la curva de aprendizaje de n8n/PDF) sin afectar el cronograma. El tiempo extra consumido en las integraciones fue compensado por la finalización temprana de otras tareas de codificación. |

|  |
| --- |
| Actividades que no has iniciado o están retrasadas:  No existen actividades de la fase de programación que se encuentren retrasadas o sin iniciar. El proyecto ha completado el 100% de la fase de desarrollo y codificación antes del plazo límite establecido en la Carta Gantt. |