

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

| **Resumen de avance proyecto APT** | Durante el desarrollo del proyecto JobFlex – Plataforma de Empleo Multiperfil, se ha avanzado de forma significativa en las etapas de diseño y desarrollo, correspondientes a la metodología tradicional en cascada.  En la actualidad, el proyecto se encuentra en la fase de desarrollo, específicamente en la finalización del modelado lógico y relacional de la base de datos, y la preparación para el inicio del backend.  Hasta la fecha, se han cumplido los siguientes hitos principales:   * **Etapa de análisis:** completada en su totalidad, con levantamiento de requerimientos funcionales y no funcionales, definición de alcance y documentación de especificaciones. * **Etapa de diseño:** ejecutada con la elaboración de la arquitectura general de la plataforma, los diagramas de interacción y el diseño de base de datos. * **Etapa de desarrollo:** iniciada, con avances notables en el frontend, donde ya se implementaron las vistas principales del candidato y parte de la estructura de la interfaz general. Posteriormente, se abordó el modelado de la base de datos, actualmente en fase de finalización para dar paso al desarrollo del backend.   Si bien el orden de ejecución de algunas tareas varió respecto al cronograma inicial (priorizando el desarrollo del frontend antes del backend), los objetivos específicos y la metodología se mantienen sin modificaciones. |
| --- | --- |
| **Objetivos** | ***(sin ajustes respecto al documento original)***  **Objetivo general:** Desarrollar JobFlex – Plataforma de Empleo Multiperfil, una aplicación web que permita almacenar múltiples perfiles profesionales, postular a ofertas laborales de manera segura y privada, y facilitar a las empresas la gestión eficiente y categorizada de postulaciones, optimizando la experiencia de ambos usuarios en el proceso de vinculación laboral.  **Objetivos específicos:**   * Diseñar la plataforma y los módulos de interacción entre candidatos y empresas. * Implementar la gestión de múltiples perfiles y CVs por candidato, garantizando privacidad y seguridad de la información. * Desarrollar un sistema de postulación con seguimiento y retroalimentación, incluyendo respuestas automáticas a los candidatos. * Crear un panel administrativo para empresas con gestión de ofertas, visualización categorizada de postulaciones y métricas para la toma de decisiones. * Aplicar buenas prácticas de usabilidad, seguridad y diseño web para asegurar una experiencia de usuario intuitiva y funcional. * Verificar y validar el correcto funcionamiento de la plataforma mediante pruebas de calidad, asegurando el cumplimiento de los requisitos funcionales y no funcionales del proyecto. |
| **Metodología** | ***(sin ajustes respecto al documento original)***  El proyecto se desarrolla bajo la metodología tradicional en cascada, que permite avanzar de forma secuencial a través de etapas definidas: análisis, diseño, desarrollo, pruebas, implementación y cierre. Cada fase se documenta y valida antes de pasar a la siguiente, asegurando orden, trazabilidad y control de calidad en el proceso. La dinámica de trabajo considera reuniones semanales de seguimiento y registro continuo en el repositorio de GitHub, garantizando transparencia y control de versiones. |
| **Evidencias de avance** | Para este informe se presentan las siguientes evidencias que respaldan el desarrollo realizado:   * **Prototipo funcional:** en entorno web, correspondiente a las vistas principales del candidato y la empresa, donde es posible visualizar la estructura base del sistema. * **Modelos lógico y relacional de la base de datos**: que reflejan el cumplimiento de la fase de diseño y la documentación formal exigida por la metodología. * **Repositorio en GitHub:** con registro de commits, versiones y archivos fuente que evidencian el progreso técnico del proyecto.   Estas evidencias permiten verificar el cumplimiento de los objetivos propuestos y demuestran el avance coherente con la planificación establecida. La calidad del proyecto se ha resguardado mediante la aplicación correcta de principios de diseño de bases de datos (tercera forma normal), revisión de consistencia entre el modelo lógico y relacional, y control de avances a través del repositorio y revisiones semanales de código. |

| **2. Monitoreo del Plan de Trabajo** |
| --- |
| Examina cuidadosamente tu plan de trabajo, enfocados 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** | **Observaciones** | **Estado de avance** | **Ajustes** |
| Proponer soluciones informáticas analizando procesos de la organización | Levantamiento de requerimientos | Reuniones presenciales y online, apuntes | 1 semana | Alison Ibarra | Puede retrasarse si no se logra consenso en funcionalidades | Completado | Sin Ajustes |
| Diseñar modelo arquitectónico de soluciones sistémicas según estándares | Diseño de arquitectura de JobFlex | Diagramas en papel o software simple, VS Code | 1 semana | Alison Ibarra e Italo Muñoz | Sirve de base para todo el desarrollo posterior | Completado | Sin Ajustes |
| Construir modelos de datos escalables según requerimientos | Modelado de base de datos | Oracle Data Modeler, Azure SQL Database / scripts SQL | 1 semana | Alison Ibarra y Juan Ignacio Marquez | La normalización es clave para evitar redundancias | En Curso | Se ajustó el orden de trabajo: se desarrolló el frontend antes de completar el modelado lógico |
| Desarrollar software con técnicas que sistematicen desarrollo y mantenimiento | Desarrollo frontend | Django (templates), HTML, CSS, JavaScript, VS Code, GitHub | 2 semanas | José Figueroa e Italo Muñoz | Requiere pruebas constantes de usabilidad | Completado | Se adelantó el desarrollo del frontend antes del backend |
| Construir programas y rutinas complejas con buenas prácticas y tecnologías vigentes | Desarrollo backend | Django con Python, VS Code, GitHub | 3 semanas | José Figueroa e Italo Muñoz | Se debe validar seguridad en formularios y autenticación | No Iniciado | Inicio postergado hasta finalización del modelado de base de datos |
| Programar consultas y rutinas en BD según requerimientos | Desarrollo de base de datos | Oracle Data Modeler, Azure SQL Database | 1 semana | Juan Ignacio Marquez | Validar consistencia de datos en pruebas | No Iniciado | Sin Ajustes |
| Transformar grandes volúmenes de datos en información útil para decisiones y mejora de procesos | Desarrollo de dashboards analíticos | Django, JavaScript (gráficas), HTML, CSS | 1 semana | Alison Ibarra e Italo Muñoz | Se debe asegurar rendimiento al manejar datos | No Iniciado | Sin Ajustes |
| Implementar soluciones integrales que automaticen u optimicen procesos de negocio | Integración de módulos | Django, GitHub, Azure | 1 semana | Italo Muñoz, José Figueroa y Juan Ignacio Marquez | Posibles errores de compatibilidad en integración | No Iniciado | Sin Ajustes |
| Gestionar proyectos informáticos, apoyando la toma de decisiones | Planificación y seguimiento | Reuniones presenciales y online, cronograma propio | Continuo (todo el proyecto) | Alison Ibarra | Importante para mantener orden en tiempos de entrega | En Curso Continuo | Sin Ajustes |
| Administrar ambientes, servicios y BD para asegurar continuidad de sistemas | Pruebas, despliegue y mantenimiento | Azure, GitHub, Visual Studio Code | 2 semanas | Juan Ignacio Marquez, Italo Muñoz y José Figueroa | Puede requerir ajustes en la nube | No Iniciado | Sin Ajustes |

| **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 que han facilitado el desarrollo:**   * **Claridad en los requerimientos:** El levantamiento detallado de requerimientos funcionales y no funcionales permitió definir con precisión las actividades y los recursos necesarios, facilitando la planificación de tareas. * **Disponibilidad de herramientas y recursos:** La utilización de plataformas como GitHub, Oracle Data Modeler y Azure SQL Database permitió un seguimiento ordenado del avance y la implementación técnica eficiente. * **Trabajo colaborativo:** La coordinación entre los integrantes del equipo permitió adelantar tareas en paralelo, como el desarrollo del frontend mientras se finalizaba el modelado de base de datos.   **Factores que han dificultado el desarrollo:**   * **Reordenamiento de tareas:** La decisión de priorizar el desarrollo del frontend antes de completar totalmente el modelado de la base de datos generó ajustes en la planificación original, implicando coordinación adicional y revisiones constantes para asegurar compatibilidad entre módulos.   + **Acción correctiva:** Se estableció un seguimiento más cercano entre los responsables de frontend y base de datos, con reuniones periódicas de revisión y control de versiones en GitHub para evitar inconsistencias. * **Dependencia de validaciones externas:** Algunas actividades, como el levantamiento de requerimientos, dependieron de la disponibilidad de información y consenso entre stakeholders, lo que pudo retrasar el inicio de ciertas tareas.   + **Acción correctiva:** Se programaron reuniones de validación anticipadas y se documentaron decisiones clave para minimizar bloqueos en etapas posteriores. * **Necesidad de pruebas y ajustes constantes:** La integración de nuevos módulos y el desarrollo de dashboards analíticos requiere revisiones periódicas para asegurar correcto funcionamiento y rendimiento.   + **Acción correctiva:** Se planificaron pruebas iterativas en entornos de desarrollo controlados, estableciendo un calendario de validación que permita identificar y corregir errores de manera temprana. |
| --- |

| ***Actividades ajustadas o eliminadas:***  **Construir modelos de datos escalables según requerimientos (Modelado de base de datos):**   * **Ajuste realizado:** Se reordenó la actividad, de modo que el desarrollo del frontend se iniciara antes de completar totalmente el modelado lógico de la base de datos. * **Justificación:** Esta decisión permitió utilizar el frontend como referencia para aplicar ingeniería inversa en el modelado de la base de datos, asegurando que todas las funcionalidades necesarias estuvieran contempladas y evitando omisiones. Esto otorgó mayor claridad y precisión al diseño de la base de datos.   **Desarrollar software con técnicas que sistematicen desarrollo y mantenimiento (Desarrollo frontend):**   * **Ajuste realizado:** Se adelantó esta actividad antes de iniciar el backend, modificando el orden original planteado en el plan de trabajo. * **Justificación:** El desarrollo del frontend como primera etapa permitió generar un prototipo funcional que sirviera de guía para el modelado de la base de datos y la posterior integración de módulos, asegurando coherencia entre los componentes del sistema.   **Construir programas y rutinas complejas con buenas prácticas y tecnologías vigentes (Desarrollo backend):**   * **Ajuste realizado:** Se postergó el inicio de esta actividad hasta que se finalice el modelado de la base de datos y se avance en el frontend. * **Justificación:** Esta postergación busca evitar inconsistencias entre la base de datos y los módulos del backend, garantizando un desarrollo más ordenado y alineado con el diseño final del sistema.   En general, los ajustes realizados fueron estratégicos y no implicaron la eliminación de actividades, sino una reordenación que permitió asegurar la coherencia funcional y técnica del proyecto. |
| --- |

| ***Actividades que no has iniciado o están retrasadas:***  En el proyecto JobFlex – Plataforma de Empleo Multiperfil, algunas actividades aún no han sido iniciadas, pero esto no se debe a retrasos, sino a que su ejecución corresponde a etapas posteriores según la planificación del proyecto y la metodología en cascada. Las actividades no iniciadas son:   * Construir programas y rutinas complejas con buenas prácticas y tecnologías vigentes (Desarrollo backend) * Programar consultas y rutinas en base de datos según requerimientos * Transformar grandes volúmenes de datos en información útil para decisiones y mejora de procesos (Desarrollo de dashboards analíticos) * Implementar soluciones integrales que automaticen u optimicen procesos de negocio (Integración de módulos) * Administrar ambientes, servicios y base de datos para asegurar continuidad de sistemas (Pruebas, despliegue y mantenimiento)   **Motivo:**  Estas actividades forman parte de etapas que dependen del desarrollo previo de otras tareas, como el frontend y el modelado de la base de datos. Su inicio está planificado de manera que se respete la secuencia lógica del proyecto y se garantice la coherencia funcional y técnica del sistema.  **Estrategias para avanzar:**   * Mantener un seguimiento constante del progreso del frontend y el modelado de base de datos para asegurar que las actividades posteriores puedan iniciarse a tiempo. * Planificar reuniones de coordinación entre los responsables de cada módulo antes de iniciar cada actividad, garantizando que todos los requerimientos estén claros y documentados. * Realizar pruebas iterativas y revisiones de consistencia de datos al iniciar cada nueva actividad, asegurando que no se produzcan errores ni incompatibilidades entre módulos.   Con esta estrategia, se asegura que el proyecto avance de manera ordenada y controlada, evitando cualquier afectación al cumplimiento de los objetivos y plazos del Proyecto APT. |
| --- |