**Evaluación Fase 1**

**Nombre: Autoevaluación Definición Proyecto APT**

**“Sistema Básico de Inventario para Pequeñas Empresas”**

**• Descripción de proyecto APT**

El objetivo de este proyecto es desarrollar un sistema de inventario que sea intuitivo, fácil de usar y accesible para los propietarios de pequeñas empresas. El sistema incluye la capacidad de registrar recibos y salidas de inventario, alertar sobre escasez de existencias y generar informes simples para facilitar la toma de decisiones. Este proyecto utiliza tecnologías web accesibles y bases de datos livianas para garantizar la economía y la eficiencia del sistema.

The goal of this project is to develop an inventory management system that is intuitive, easy to use, and accessible for small business owners. The system includes the capability to log inventory receipts and outflows, alert on stock shortages, and generate simple reports to facilitate decision making. This project utilizes accessible web technologies and lightweight databases to ensure the system's affordability and efficiency.

**• Relación del proyecto APT con las competencias del perfil de egreso.**

Este proyecto está directamente relacionado con las competencias de desarrollo de software, análisis de sistemas y gestión de bases de datos que forman parte del perfil de egreso de la carrera de informática.

This project is directly related to the software development, systems analysis, and database management competencies that are part of the graduate profile for the computer science degree program.

**• Relación del proyecto con tus intereses profesionales.**

Desarrollar este proyecto alinea con mis intereses profesionales en tecnología de la información y desarrollo de software, especialmente en contextos que impactan directamente el rendimiento y crecimiento empresarial. Este proyecto ofrece la oportunidad de explorar y profundizar en la gestión de proyectos tecnológicos aplicados al mundo empresarial.

Developing this project aligns with my professional interests in information technology and software development, especially in contexts that directly impact business performance and growth. This project offers the opportunity to explore and delve deeper into the management of technology projects applied to the business world.

**• Argumento del por qué el proyecto es factible a realizarse dentro de la asignatura.**

El proyecto "Sistema Básico de Inventario para Pequeñas Empresas" es factible de realizar dentro de la asignatura debido a que está diseñado para completarse en un semestre, aprovechando las horas asignadas y utilizando tecnologías y herramientas ya disponibles, como Python o Java, y bases de datos accesibles. Además, cuentas con competencias previas en programación, análisis de sistemas y gestión de bases de datos, lo que facilita su desarrollo. La colaboración con pequeñas empresas locales para la prueba del sistema es posible y manejable, y los posibles desafíos externos, como el tiempo limitado, son controlables dentro del contexto del curso.

The "Basic Inventory System for Small Businesses" project is feasible to carry out within the course because it is designed to be completed in one semester, taking advantage of the assigned hours and using already available technologies and tools, such as Python or Java, and accessible databases. Additionally, you have prior competencies in programming, systems analysis, and database management, which facilitates its development. Collaborating with local small businesses for system testing is possible and manageable, and potential external challenges, such as limited time, are controllable within the context of the course.

**• Objetivos claros y coherentes.**

Objetivo Principal:

* Desarrollar un sistema básico de inventario que sea accesible y fácil de usar para pequeñas empresas, permitiendo una gestión eficiente de sus recursos y mejorando la toma de decisiones mediante el control automatizado de entradas y salidas de stock.

Objetivos Secundarios:

1. Implementar un sistema de alertas que notifique automáticamente sobre la escasez de inventario para prevenir faltantes.
2. Desarrollar un módulo de generación de informes simples que permita a los propietarios de pequeñas empresas analizar el estado de su inventario de manera clara y rápida.
3. Aplicar conocimientos de programación y gestión de bases de datos para crear un sistema liviano y eficiente que pueda ser fácilmente implementado sin requerir hardware o software costosos.
4. Probar y validar el sistema en colaboración con pequeñas empresas locales para asegurar que se adapte a las necesidades reales del usuario final.

Main Objective:

* Develop a basic inventory system that is accessible and easy to use for small businesses, enabling efficient management of their resources and improving decision-making through automated tracking of stock inflows and outflows.

Secondary Objectives:

1. Implement an alert system that automatically notifies about inventory shortages to prevent stockouts.
2. Develop a module for generating simple reports that allows small business owners to quickly and clearly analyze the state of their inventory.
3. Apply programming and database management knowledge to create a lightweight and efficient system that can be easily implemented without requiring expensive hardware or software.
4. Test and validate the system in collaboration with local small businesses to ensure it adapts to the real needs of the end-user.

**• Propuesta metodológica de trabajo que permita alcanzar los objetivos.**

Propuesta Metodológica de Trabajo:

Para alcanzar los objetivos del proyecto "Sistema Básico de Inventario para Pequeñas Empresas," se propone utilizar una metodología ágil, que permita un desarrollo iterativo e incremental, asegurando la adaptabilidad y la entrega continua de valor. El proceso se estructurará en las siguientes fases:

1. Fase de Requisitos y Análisis:
   * Realizar reuniones iniciales con los stakeholders (pequeñas empresas colaboradoras) para identificar las necesidades específicas del sistema de inventario.
   * Definir los requisitos funcionales y no funcionales del sistema, priorizando las funcionalidades esenciales como el registro de inventario, generación de informes y alertas automáticas.
   * Crear un backlog de tareas basado en estos requisitos, priorizando las más críticas para el negocio.
2. Fase de Diseño:
   * Diseñar la arquitectura del sistema, enfocándose en la modularidad y escalabilidad, para facilitar futuras expansiones o modificaciones.
   * Elaborar wireframes y prototipos de las interfaces de usuario para asegurar una experiencia intuitiva y fácil de usar, validando estos prototipos con los stakeholders.
3. Fase de Desarrollo:
   * Implementar el sistema de inventario en ciclos de desarrollo cortos (sprints), con entregas parciales y funcionales al final de cada sprint.
   * Priorizar el desarrollo de funcionalidades clave como la gestión de inventarios, la generación de alertas, y los módulos de informes.
   * Realizar pruebas unitarias y de integración continuas para asegurar la calidad y funcionalidad del sistema conforme se va desarrollando.
4. Fase de Pruebas y Validación:
   * Desplegar versiones de prueba del sistema en entornos controlados dentro de las empresas colaboradoras para obtener feedback real sobre su funcionamiento.
   * Realizar pruebas de aceptación con los usuarios finales para validar que el sistema cumple con los requisitos establecidos y ajustarlo según las observaciones recibidas.
5. Fase de Implementación y Documentación:
   * Finalizar el desarrollo con una implementación completa del sistema en el entorno de las pequeñas empresas colaboradoras.
   * Documentar el sistema, incluyendo guías de usuario, manuales técnicos, y un informe final que detalle el proceso de desarrollo, los resultados obtenidos, y las lecciones aprendidas.
6. Fase de Seguimiento y Soporte:
   * Ofrecer soporte post-implementación para resolver posibles problemas y realizar ajustes menores según las necesidades del usuario final.
   * Evaluar el desempeño del sistema después de su implementación para identificar oportunidades de mejora o expansión futura.

Methodological-Proposal:  
To achieve the objectives of the "Basic Inventory System for Small Businesses" project, an agile methodology is proposed, which will allow for an iterative and incremental development, ensuring adaptability and continuous delivery of value. The process will be structured in the following phases:

1. Requirements and Analysis Phase:
   * Hold initial meetings with stakeholders (collaborating small businesses) to identify the specific needs of the inventory system.
   * Define the functional and non-functional requirements of the system, prioritizing essential functionalities such as inventory tracking, reporting, and automatic alerts.
   * Create a task backlog based on these requirements, prioritizing the most critical ones for the business.
2. Design Phase:
   * Design the system architecture, focusing on modularity and scalability, to facilitate future expansions or modifications.
   * Develop wireframes and prototypes of the user interfaces to ensure an intuitive and easy-to-use experience, validating these prototypes with stakeholders.
3. Development Phase:
   * Implement the inventory system in short development cycles (sprints), with partial and functional deliveries at the end of each sprint.
   * Prioritize the development of key functionalities such as inventory management, alert generation, and reporting modules.
   * Perform continuous unit and integration testing to ensure the quality and functionality of the system as it is being developed.
4. Testing and Validation Phase:
   * Deploy test versions of the system in controlled environments within the collaborating companies to obtain real feedback on its performance.
   * Conduct acceptance tests with end-users to validate that the system meets the established requirements and adjust it according to the feedback received.
5. Implementation and Documentation Phase:
   * Complete the development with a full implementation of the system in the environment of the collaborating small businesses.
   * Document the system, including user guides, technical manuals, and a final report detailing the development process, the results obtained, and the lessons learned.
6. Monitoring and Support Phase:
   * Offer post-implementation support to resolve possible problems and make minor adjustments based on the end-user's needs.
   * Evaluate the system's performance after its implementation to identify opportunities for improvement or future expansion.

**• Plan de trabajo para el proyecto APT.**

**Fase 1: Iniciación**

* Actividades: Definir el alcance, identificar stakeholders, reunión inicial.
* Duración: 1 semana.
* Recursos: Tiempo del estudiante, herramientas de comunicación (Zoom, correo electrónico).
* Facilitadores: Acceso fácil a herramientas de comunicación, claridad en los objetivos iniciales.
* Obstáculos: Posible dificultad en la coordinación de horarios con los stakeholders.

**Fase 2: Requisitos y Análisis**

* Actividades: Recolección de requisitos, especificaciones, priorización de funcionalidades.
* Duración: 2 semanas.
* Recursos: Entrevistas con stakeholders, herramientas de documentación (Word, Excel).
* Facilitadores: Disponibilidad de herramientas para documentar requisitos, experiencia previa en análisis de sistemas.
* Obstáculos: Complejidad en entender todas las necesidades de los usuarios finales, resistencia al cambio por parte de los stakeholders.

**Fase 3: Diseño**

* Actividades: Diseño de la arquitectura del sistema, creación de wireframes y prototipos, plan de pruebas preliminar.
* Duración: 2 semanas.
* Recursos: Software de diseño (Figma, Lucidchart), herramientas de documentación.
* Facilitadores: Acceso a software de diseño, conocimientos previos en diseño de sistemas.
* Obstáculos: Limitaciones en la experiencia de diseño de interfaces, posible falta de claridad en los requisitos que afecte el diseño.

**Fase 4: Desarrollo**

* Actividades: Implementación del sistema en 3 sprints, pruebas unitarias y de integración.
* Duración: 6 semanas.
* Recursos: Entorno de desarrollo (IDE, repositorios de código), base de datos.
* Facilitadores: Conocimientos en programación, modularidad del sistema.
* Obstáculos: Posible falta de tiempo para completar todo el desarrollo en un sprint, desafíos técnicos imprevistos.

**Fase 5: Pruebas y Validación**

* Actividades: Pruebas de aceptación, validación del sistema, ajustes según feedback.
* Duración: 2 semanas.
* Recursos: Feedback de usuarios finales, entorno de pruebas.
* Facilitadores: Acceso a usuarios finales para pruebas, estructura de pruebas definida.
* Obstáculos: Retrasos en la obtención de feedback, posibles errores no detectados en fases previas.

**Fase 6: Implementación**

* Actividades: Despliegue del sistema, capacitación de usuarios, documentación.
* Duración: 1 semana.
* Recursos: Manuales de usuario, tiempo para capacitación.
* Facilitadores: Estructura simple del sistema que facilita la implementación, manuales de usuario claros.
* Obstáculos: Resistencia al cambio por parte de los usuarios finales, posibles problemas técnicos durante la implementación.

**Fase 7: Seguimiento y Cierre**

* Actividades: Monitoreo del sistema, soporte, reunión de cierre, documentación final.
* Duración: 1 semana.
* Recursos: Tiempo para soporte, herramientas de comunicación.
* Facilitadores: Experiencia previa en soporte técnico, relaciones establecidas con los stakeholders.
* Obstáculos: Tiempo limitado para monitorear el sistema, posibles solicitudes de ajustes fuera del alcance inicial.

**Phase 1: Initiation**

• Activities: Define the scope, identify stakeholders, initial meeting.

• Duration: 1 week.

• Resources: Student's time, communication tools (Zoom, email).

• Facilitators: Easy access to communication tools, clarity on initial objectives.

• Obstacles: Possible difficulty in coordinating schedules with stakeholders.

**Phase 2: Requirements and Analysis**

• Activities: Requirements gathering, specifications, prioritization of functionalities.

• Duration: 2 weeks.

• Resources: Stakeholder interviews, documentation tools (Word, Excel).

• Facilitators: Availability of tools to document requirements, previous experience in system analysis.

• Obstacles: Complexity in understanding all the needs of end users, resistance to change from stakeholders.

**Phase 3: Design**

• Activities: Design of the system architecture, creation of wireframes and prototypes, preliminary test plan.

• Duration: 2 weeks.

• Resources: Design software (Figma, Lucidchart), documentation tools.

• Facilitators: Access to design software, prior knowledge in system design.

• Obstacles: Limitations in interface design experience, possible lack of clarity in requirements affecting the design.

**Phase 4: Development**

• Activities: Implementation of the system in 3 sprints, unit and integration testing.

• Duration: 6 weeks.

• Resources: Development environment (IDE, code repositories), database.

• Facilitators: Programming knowledge, modularity of the system.

• Obstacles: Possible lack of time to complete all development in a sprint, unforeseen technical challenges.

**Phase 5: Testing and Validation**

• Activities: Acceptance testing, system validation, adjustments based on feedback.

• Duration: 2 weeks.

• Resources: Feedback from end-users, testing environment.

• Facilitators: Access to end-users for testing, defined test structure.

• Obstacles: Delays in obtaining feedback, possible errors not detected in previous phases.

**Phase 6: Implementation**

• Activities: System deployment, user training, documentation.

• Duration: 1 week.

• Resources: User manuals, time for training.

• Facilitators: Simple system structure that facilitates implementation, clear user manuals.

• Obstacles: Resistance to change from end-users, possible technical problems during implementation.

**Phase 7: Monitoring and Closure**

• Activities: System monitoring, support, closure meeting, final documentation.

• Duration: 1 week.

• Resources: Time for support, communication tools.

• Facilitators: Previous experience in technical support, established relationships with stakeholders.

• Obstacles: Limited time to monitor the system, possible requests for adjustments outside the initial scope.

**• Propuesta de evidencias que darán cuenta del logro de las actividades.**

* **Fase de Iniciación:**
* Evidencias:
  + Documento de Definición de Alcance y Objetivos: Un documento detallado que define claramente el alcance del proyecto, los objetivos principales y secundarios, y la identificación de los stakeholders.
  + Lista de Stakeholders y Plan de Comunicación: Una lista detallada de los stakeholders clave y un plan que describa cómo y cuándo se les comunicará el progreso del proyecto.
* **Fase de Requisitos y Análisis:**
* Evidencias:
  + Documento de Especificaciones de Requisitos: Un documento que recopile los requisitos funcionales y no funcionales, incluyendo diagramas de casos de uso y escenarios de usuario.
  + Backlog Inicial de Tareas: Una lista priorizada de las funcionalidades y tareas a realizar, basada en los requisitos identificados.
* **Fase de Diseño:**
* Evidencias:
  + Diagramas de Arquitectura del Sistema: Diagramas que muestren la estructura del sistema, incluyendo la arquitectura de software y la base de datos.
  + Prototipos y Wireframes: Mockups o prototipos de la interfaz de usuario, validados con los stakeholders.
  + Documento del Plan de Pruebas Preliminar: Un documento que describa las pruebas planeadas para cada módulo del sistema.
* **Fase de Desarrollo:**
* Evidencias:
  + Código Fuente Documentado: El código fuente del sistema, alojado en un repositorio con control de versiones, incluyendo comentarios y documentación técnica.
  + Informe de Pruebas Unitarias y de Integración: Resultados de las pruebas realizadas durante el desarrollo, documentando los casos de prueba y los resultados obtenidos en cada sprint.
  + Versiones de Prueba del Sistema: Versiones funcionales del sistema entregadas al final de cada sprint, demostrando el avance en el desarrollo.
* **Fase de Pruebas y Validación:**
* Evidencias:
  + Informe de Resultados de las Pruebas de Aceptación: Un informe detallado que recoja el feedback de los usuarios finales y describa cómo se han abordado los problemas detectados.
  + Versiones Ajustadas del Sistema: La versión final del sistema, ajustada según el feedback recibido y validada por los usuarios finales.
* **Fase de Implementación:**
* Evidencias:
  + Sistema Implementado y Operativo: Evidencia de que el sistema ha sido desplegado con éxito en las empresas colaboradoras, acompañado de capturas de pantalla o demostraciones en video.
  + Manuales de Usuario y Documentación Técnica: Documentación completa que incluya guías de usuario y manuales técnicos para el mantenimiento del sistema.
* **Fase de Seguimiento y Cierre:**
* Evidencias:
  + Informe de Monitoreo y Soporte: Un informe que detalle las actividades de soporte y cualquier ajuste realizado durante el periodo de seguimiento.
  + Reunión Final y Evaluación del Proyecto: Minutas de la reunión de cierre con los stakeholders, incluyendo su evaluación del éxito del proyecto y lecciones aprendidas.
  + Informe de Lecciones Aprendidas y Cierre Formal del Proyecto: Un documento final que resuma las lecciones aprendidas durante el proyecto y formalice su cierre.
* **Initiation Phase:**

• Deliverables:

o Scope and Objectives Definition Document: A detailed document that clearly defines the project scope, the main and secondary objectives, and the identification of stakeholders.

o Stakeholder List and Communication Plan: A detailed list of key stakeholders and a plan that describes how and when they will be communicated the project progress.

• **Requirements and Analysis Phase:**

• Deliverables:

o Requirements Specification Document: A document that compiles the functional and non-functional requirements, including use case diagrams and user scenarios.

o Initial Task Backlog: A prioritized list of functionalities and tasks to be performed, based on the identified requirements.

**• Design Phase:**

• Deliverables:

o System Architecture Diagrams: Diagrams showing the structure of the system, including software architecture and the database.

o Prototypes and Wireframes: UI mockups or prototypes, validated with the stakeholders.

o Preliminary Test Plan Document: A document that describes the planned tests for each system module.

**• Development Phase:**

• Deliverables:

o Documented Source Code: The system's source code, hosted in a version-controlled repository, including comments and technical documentation.

o Unit and Integration Test Report: Results of the tests performed during development, documenting the test cases and the results obtained in each sprint.

o System Proof-of-Concept Versions: Functional versions of the system delivered at the end of each sprint, demonstrating the progress in development.

**• Testing and Validation Phase:**

• Deliverables:

o Acceptance Test Results Report: A detailed report that collects the feedback from end-users and describes how the detected issues have been addressed.

o Adjusted System Versions: The final version of the system, adjusted according to the received feedback and validated by the end-users.

**• Implementation Phase:**

• Deliverables:

o Implemented and Operational System: Evidence that the system has been successfully deployed in the collaborating companies, accompanied by screenshots or video demonstrations.

o User Manuals and Technical Documentation: Complete documentation including user guides and technical manuals for system maintenance.

• **Monitoring and Closure Phase:**

• Deliverables:

o Monitoring and Support Report: A report detailing the support activities and any adjustments made during the monitoring period.

o Final Meeting and Project Evaluation: Minutes of the closing meeting with stakeholders, including their evaluation of the project's success and lessons learned.

o Lessons Learned Report and Formal Project Closure: A final document that summarizes the lessons learned during the project and formalizes its closure.