

Título del Proyecto: Simulador de Inventario para Almacenes

1. Resumen Ejecutivo

Este proyecto consiste en el desarrollo de un simulador de inventario para almacenes, utilizando ensamblador ARM64 y el lenguaje C. La aplicación simula la gestión de productos en un almacén con funcionalidades de entrada y salida de inventario, actualización de existencias y generación de reportes. El objetivo es crear un sistema que permita simular en tiempo real el flujo de inventario en un entorno de bajo nivel, aprovechando la eficiencia del ensamblador ARM64 combinado con la flexibilidad del lenguaje C.

Problema que aborda: La necesidad de gestionar inventarios en tiempo real de forma eficiente, con un sistema que permita verificar entradas, salidas y existencias, todo en un entorno de bajo nivel que optimice el uso de recursos.

Solución a desarrollar: Un sistema de inventario por comandos que permite registrar entradas y salidas de productos en una interfaz de línea de comandos. La aplicación también ofrecerá reportes, calculará cantidades disponibles y emitirá advertencias cuando el inventario esté bajo o excedido.

3. Objetivos del Aprendizaje

3.1 Objetivo General

Que el estudiante aplique conocimientos de ensamblador y C en la creación de un sistema de simulación de inventario, comprendiendo el manejo de memoria, estructuras de datos y algoritmos eficientes.

3.2 Objetivos Específicos

Al finalizar el proyecto, los estudiantes deberán ser capaces de:

- 1. Implementar operaciones aritméticas y de control de flujo en ensamblador.**
- 2. Manejar estructuras de datos básicas en C para organizar el inventario.**
- 3. Desarrollar un sistema de comandos en línea para la interacción con el usuario.**

4. Enunciado del Proyecto

3.1 Descripción del problema a resolver

El proyecto simulará un sistema de inventario donde el usuario puede registrar productos, actualizar existencias y consultar el estado del inventario. Se deberá

interactuar con el sistema a través de comandos en una interfaz de línea de comandos, lo cual permite administrar y verificar en tiempo real el flujo de inventario.

3.2 Alcance del proyecto

- **Alcance obligatorio:** Implementar el registro de productos y las operaciones de entrada y salida de inventario.
- **Alcance opcional:** Desarrollar reportes detallados y alertas personalizables según el nivel de inventario.

3.3 Requerimientos técnicos

- **Uso de ARM64 en un entorno de emulación (QEMU o Raspberry Pi) para ejecutar el código ensamblador.**
- **Integración de funciones en C para manejar estructuras de datos.**

4.4 Entregables

Describe los productos concretos que se espera que los estudiantes entreguen al finalizar el proyecto. Pueden ser prototipos, informes técnicos, documentación, etc.

Ejemplo:

Tipo	Descripción
Documentación Técnica	Incluye manuales de usuario, guías de instalación, diagramas y descripciones de la arquitectura o flujo del sistema desarrollado.
Código Fuente	El código completo del proyecto, con una estructura clara y comentada, que refleja las buenas prácticas de programación y uso de control de versiones.
Manual de Usuario	Documento que explica cómo usar el sistema desarrollado, incluyendo capturas de pantalla, pasos detallados y resolución de problemas comunes.

Diagrama de Flujo	Representación gráfica del flujo de trabajo o funcionamiento del sistema, que permite comprender el recorrido de los datos o acciones dentro del proyecto.
Informe de Impacto	Documento que describe el impacto potencial del proyecto en su contexto de aplicación, como mejoras de eficiencia, reducción de costos, etc.
Plan de Mantenimiento	Propuesta de cómo se mantendrá y actualizará el sistema a largo plazo, incluyendo posibles mejoras o futuros desarrollos.

- **Prototipo funcional de la aplicación:** Se debe entregar un archivo APK funcional de la aplicación móvil, junto con el código fuente.
- **Informe técnico:** Un informe de no más de 10 páginas que describa el proceso de desarrollo, los retos encontrados y cómo se resolvieron.
- **Documentación del usuario:** Manual de uso de la aplicación con capturas de pantalla.

5. Metodología

Fase 1: Planificación

- **Analizar el enunciado** y definir las operaciones necesarias.
- **Dividir en módulos** (registro de productos, operaciones de inventario, generación de reportes).
- **Cronograma** para cada módulo, asignando tiempos para programación, pruebas y documentación.

Fase 2: Diseño

- **Estructura de datos en C:** Planificar cómo se almacenará cada producto (e.g., en structs).
- **Definir la lógica de cada comando** en ensamblador (e.g., comandos de entrada/salida de inventario).
- **Formato de comandos:** Planificar sintaxis (e.g., ENTRAR [Producto] [Cantidad], SALIR [Producto] [Cantidad]).

Fase 3: Implementación

- **Programación modular:** Crear funciones básicas en ensamblador para cada comando.
- **Manejo de errores:** Validaciones para cantidades negativas o excedentes.
- **Integración en C:** Implementar estructuras y manejo de listas de inventario.

Fase 4: Pruebas y Documentación

- **Pruebas unitarias:** Verificar la funcionalidad de cada comando.
- **Pruebas de integración:** Ejecutar la aplicación completa para confirmar el flujo de inventario.
- **Documentación:** Crear manuales técnicos y de usuario.

6. Desarrollo de Habilidades Blandas

Para complementar el desarrollo técnico, esta sección se centra en las habilidades blandas que los estudiantes deberán mejorar a lo largo del proyecto, como la comunicación, el liderazgo, y la colaboración en equipo.

6.2 Proyectos Individuales

Los proyectos individuales permiten a los estudiantes desarrollar autonomía y responsabilidad sobre su propio trabajo. En este tipo de proyectos, cada estudiante debe gestionar su tiempo, investigar, planificar y ejecutar todas las fases del proyecto de manera independiente.

6.2.1 Autogestión del Tiempo

Los estudiantes deben crear un cronograma personal para cumplir con los plazos establecidos. Esto les ayuda a mejorar su disciplina y capacidad de priorización, habilidades esenciales en cualquier entorno profesional.

6.2.2 Responsabilidad y Compromiso

En un proyecto individual, el estudiante asume la totalidad de las responsabilidades, desde la investigación hasta la entrega final. Esto fomenta el sentido de compromiso y permite una mayor personalización en la solución del problema planteado.

6.2.3 Resolución de Problemas

Trabajar de manera independiente impulsa a los estudiantes a buscar soluciones de forma creativa y a enfrentarse a los desafíos sin depender del apoyo constante de otros. Esto fortalece su capacidad para resolver problemas de manera autónoma.

6.2.4 Reflexión Personal

Al concluir el proyecto, el estudiante realiza una autoevaluación, reflexionando sobre sus

decisiones, lo aprendido y las áreas en las que podría mejorar. Esta práctica promueve el desarrollo continuo y el autoaprendizaje.

7. Cronograma

El cronograma describe las etapas clave del proyecto, los plazos estimados para cada una, y el proceso de asignación, elaboración y calificación de las tareas. Los estudiantes deberán seguir este plan para asegurar que el proyecto avance de manera organizada y cumpla con los plazos establecidos. Cada fase incluye la asignación de tareas, el tiempo estimado para su elaboración, y el momento de su calificación.

Tipo	Fecha Inicio	Fecha Fin
Asignación de Proyecto	Por definir	Por definir
Elaboración	Por definir	Por definir
Calificación	Por definir	Por definir

8. Evaluación

La evaluación del proyecto se realizará en función de varios criterios clave, teniendo en cuenta tanto los aspectos técnicos como las habilidades blandas demostradas a lo largo del desarrollo. A continuación, se presenta un **ejemplo** de cómo podría distribuirse la puntuación, aunque esto puede cambiar según sea necesario y las particularidades del proyecto.

Resumen de Puntuaciones

Área	Puntos Totales	Puntos Obtenidos
Funcionalidad del Proyecto	80	
Procedimiento y Desarrollo	10	
Innovación y Originalidad	5	

Habilidades Blandas y Presentación	5	
Total	100	

Detalle de la Calificación

Criterio	Descripción	Puntos Máximos	Puntuación Obtenida
1. Funcionalidad del Proyecto		80	
1.1 Cumplimiento de Requisitos	Grado en que el proyecto cumple con las especificaciones y requisitos establecidos.		
1.2 Funcionamiento en Pruebas	¿El proyecto funciona correctamente bajo condiciones normales y de prueba?		
1.3 Eficiencia del Código	Calidad y eficiencia en la implementación del código, incluyendo su organización y claridad.		
2. Procedimiento y Desarrollo		10	
2.1 Metodología de Desarrollo	¿Se utilizó una metodología clara y organizada durante el desarrollo del proyecto?		
2.2 Solución de Problemas	Capacidad para identificar, documentar y resolver problemas técnicos durante el desarrollo.		
2.3 Documentación	Claridad y detalle en la documentación presentada del proyecto.		

3. Innovación y Originalidad		5	
3.1 Creatividad en la Solución	Grado de innovación y originalidad en las soluciones técnicas o creativas implementadas.		
3.2 Extensión Más Allá de los Requisitos	¿El proyecto fue más allá de los requerimientos mínimos en términos de funcionalidad o enfoque?		
4. Habilidades Blandas y Presentación		5	
4.1 Trabajo en Equipo	(Para proyectos grupales) Colaboración efectiva entre los miembros del equipo.		
4.2 Comunicación y Presentación	Claridad en la presentación de los resultados, organización de ideas, y exposición.		
Total de Puntos		100	

Comentarios Generales: