

Presentación del proyecto: Creación de un lenguaje de programación.

Fase 1: Planificación y Diseño

1. Definición de Objetivos:

- Especifica claramente los objetivos y características que deseas lograr con tu lenguaje.

2. Estudio de Viabilidad:

- Evalúa la viabilidad y los recursos necesarios para el proyecto.

3. Investigación:

- Investiga sobre lenguajes de programación existentes y conceptos teóricos relevantes.

4. Diseño Preliminar:

- Diseña la sintaxis inicial y las características clave del lenguaje.

Fase 2: Implementación del Compilador/Intérprete

5. Especificación Formal:

- Detalla la sintaxis y semántica del lenguaje en una especificación formal.

6. Implementación del Analizador Léxico y Sintáctico:

- Desarrolla el analizador léxico y sintáctico para procesar el código fuente.

7. Análisis Semántico y Generación de Código Intermedio:

- Implementa el análisis semántico y la generación de código intermedio.

8. Generación de Código Objeto y Optimización (opcional):

- Desarrolla el generador de código objeto y aplica técnicas básicas de optimización.

Fase 3: Implementación del Intérprete y Pruebas

9. Construcción del Intérprete:

- Implementa un intérprete para ejecutar el código fuente directamente.

10. Pruebas y Depuración: - Realiza pruebas exhaustivas para garantizar la corrección del compilador/intérprete.

Fase 4: Refinamiento y Documentación

11. Mejoras Finales: - Realiza mejoras en el compilador/intérprete según la retroalimentación y la evaluación.

12. Documentación: - Crea una documentación completa que explique la sintaxis, semántica y uso del lenguaje.

Fase 5: Presentación y Evaluación

13. Preparación de la Presentación: - Organiza una presentación para mostrar tu proyecto.

14. Evaluación: - Evalúa el proyecto en términos de diseño, implementación, documentación y presentación.

Consejos Prácticos:

- **División de Tareas:** Divide el proyecto en tareas más pequeñas y manejables para facilitar el progreso.
- **Iteración:** No tengas miedo de iterar sobre el diseño y la implementación a medida que avanzas.
- **Priorización:** Prioriza las características clave y asegúrate de tener un producto funcional antes de agregar características más avanzadas.
- **Colaboración:** Si es un proyecto en equipo, comunica eficientemente y colabora en el diseño e implementación.
- **Uso de Herramientas:** Utiliza herramientas y bibliotecas existentes cuando sea posible para acelerar el desarrollo.

Requerimientos

Requisitos Funcionales:

1. **Sintaxis Clara:**
 - El lenguaje debe tener una sintaxis clara y fácil de entender para los programadores.
2. **Tipos de Datos:**
 - El lenguaje debe admitir una variedad de tipos de datos, como enteros, decimales, cadenas, booleanos, etc.
3. **Estructuras de Control:**
 - Debe proporcionar estructuras de control de flujo, como condicionales (if-else) y bucles (for, while).
4. **Funciones y Procedimientos:**
 - Debe permitir la definición y llamada de funciones y procedimientos.
5. **Entrada/Salida:**
 - Debe admitir operaciones de entrada y salida para interactuar con el usuario y otros programas.
6. **Manejo de Errores:**
 - Debe tener un sistema robusto de manejo de errores, proporcionando mensajes descriptivos y recuperación adecuada.
7. **Operaciones Aritméticas y Lógicas:**
 - Debe permitir operaciones aritméticas y lógicas básicas.

Requisitos No Funcionales:

1. **Eficiencia:**
 - El lenguaje debe ser eficiente en términos de tiempo de ejecución y consumo de recursos.
2. **Documentación:**
 - Debe proporcionar documentación clara y completa para programadores, incluyendo una referencia del lenguaje y ejemplos.