



Trabajo Fin de Grado

flowming →

Aplicación web para el diseño y
ejecución de diagramas de flujo

Autor

Daniel Pérez Fernández

[Redacted]

Tutor

Álvaro Montero Montes

Índice

1. Problema
2. Análisis
3. Solución
4. Validación
5. Conclusiones

1

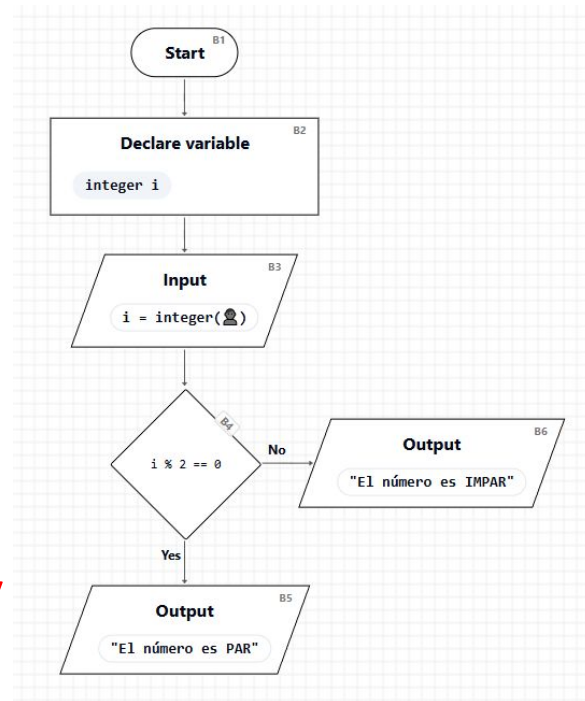
Problema

Java

```
public class ParImpar {  
    public static void main(String[] args) {  
        Scanner teclado = new Scanner(System.in);  
  
        System.out.print("Introduce un número entero: ");  
        int i = teclado.nextInt();  
  
        if (i % 2 == 0) {  
            System.out.println("El número es PAR");  
        } else {  
            System.out.println("El número es IMPAR");  
        }  
    }  
}
```

¡Operador incorrecto!
¡Falta un punto y coma!

Diagrama de flujo (Flowming)



1

Problema

Java

```
public class ParImpar {  
    public static void main(String[] args) {  
        Scanner teclado = new Scanner(System.in);  
  
        System.out.print("Introduce un número entero: ");  
        int i = teclado.nextInt();  
  
        if (i % 2 == 0) {  
            System.out.println("El número es PAR");  
        } else {  
            System.out.println("El número es IMPAR");  
        }  
    }  
}
```

C++

```
int main() {  
    int i;  
    std::cout << "Introduce un número entero: ";  
    std::cin >> i;  
  
    if (i % 2 == 0) {  
        std::cout << "El número es PAR" << std::endl;  
    } else {  
        std::cout << "El número es IMPAR" << std::endl;  
    }  
  
    return 0;  
}
```

JavaScript

```
const i_str = prompt("Introduce un número entero: ");  
const i = parseInt(i_str, 10);  
  
if (i % 2 == 0) {  
    console.log("El número es PAR");  
} else {  
    console.log("El número es IMPAR");  
}
```

Python

```
i_str = input("Introduce un número entero")  
i = int(i_str)  
  
if i % 2 == 0:  
    print("El número es PAR")  
else:  
    print("El número es IMPAR")
```



1. Programación por Bloques

Alice 




Blockly

- Plataforma: Web/Escritorio



Muy Accesible (🌐)



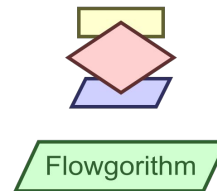
Muy Intuitivo (🧩)



Transición Discutida (↔)

(Weintrop, 2019)

2. Diagramas de Flujo



- Plataforma: Escritorio



Poco Accesible (📦)



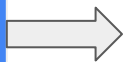
Fuerte Base Algorítmica (🧠)



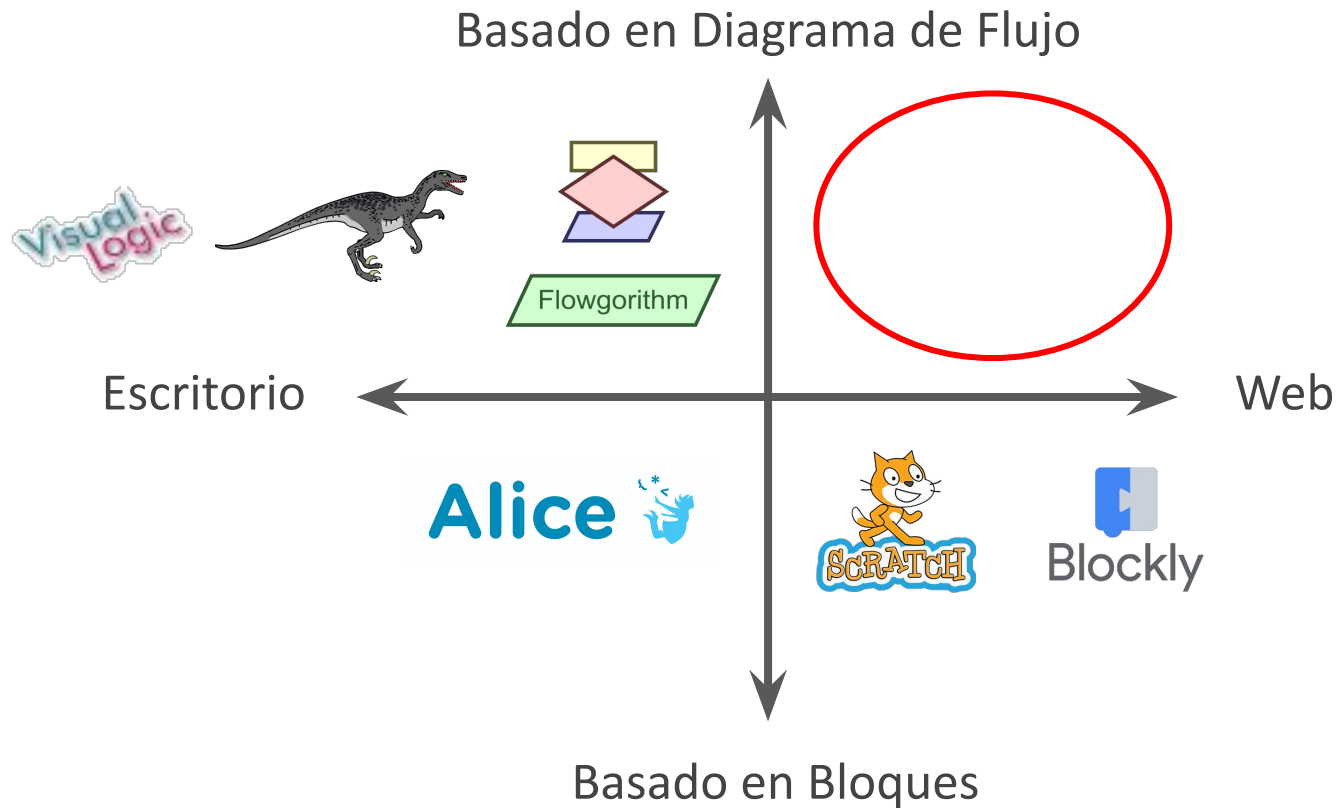
Salto Más Directo (🏰)

2

Análisis



La oportunidad





Suavizar la curva de aprendizaje de la programación



Interfaz Web Intuitiva

Crear un entorno para diseñar diagramas de forma **sencilla** y **visual**, eliminando la fricción del usuario.



Motor de Ejecución

Implementar un motor que **ejecute** y **depure** los algoritmos **visualmente**, mostrando el flujo y las variables en tiempo real.



Transición a Texto

Generar **código de Python** equivalente para conectar la lógica visual con la sintaxis de un lenguaje real, facilitando la **transición** a la programación textual.

3

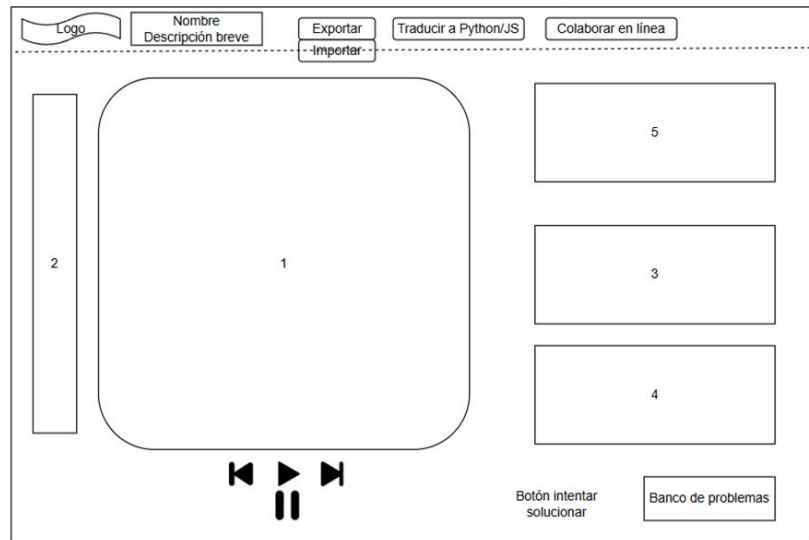
Solución



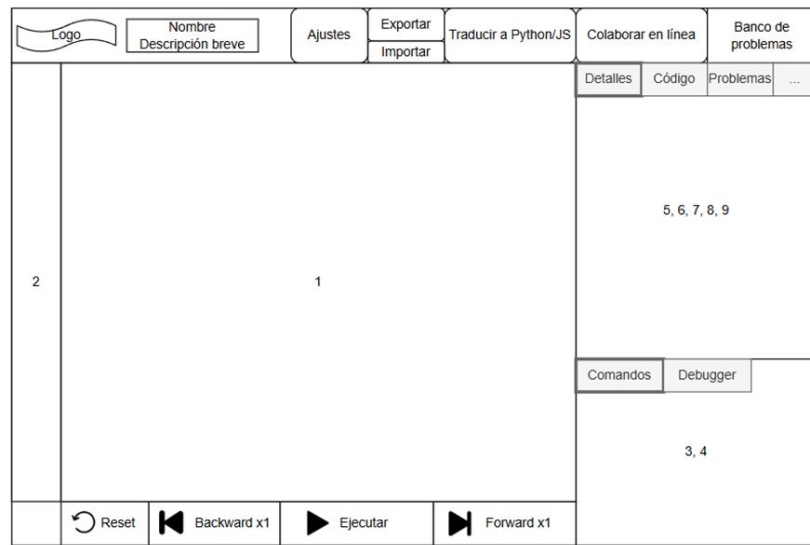
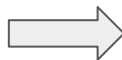
Diseño de la solución



Bocetos



Boceto inicial



Mockup detallado

3

Solución



Diseño de la solución



Logotipo

flowming →



flowming →

Boceto inicial (manual)

Versión final (IA + refinamiento)



- **Aplicación Frontend (Cliente)**

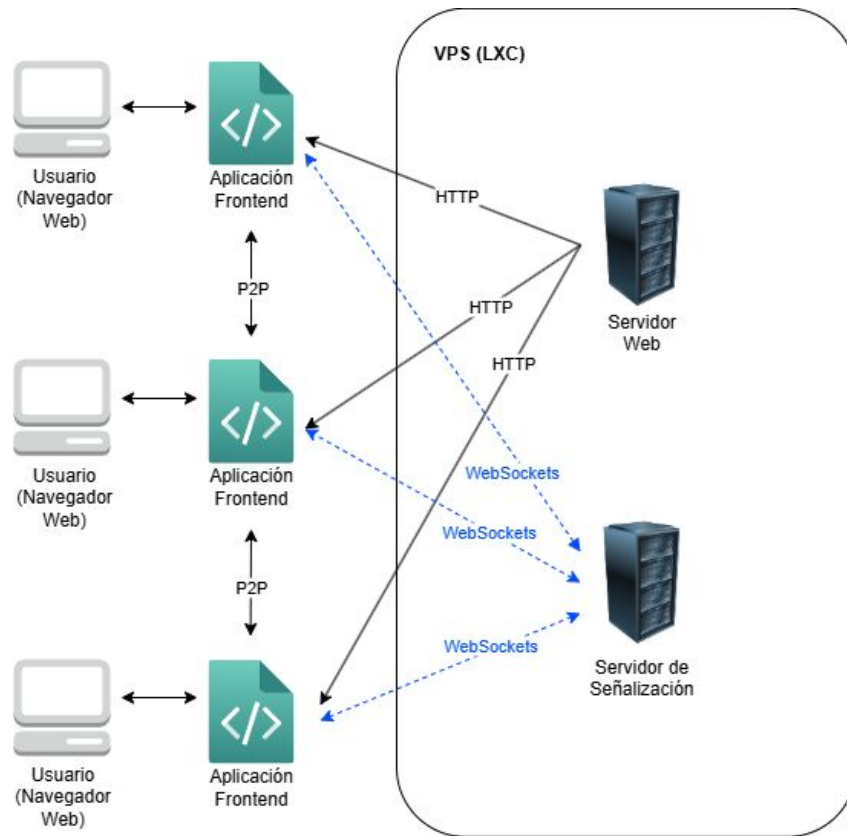
Un SPA cuya lógica se ejecuta 100% en el navegador.

- **Servidor Web (VPS)**

Un servidor **muy ligero** cuya función es servir la aplicación web a través de **HTTP**.

- **Servidor de Señalización (VPS)**

Para habilitar la colaboración **P2P** en tiempo real usando **WebSockets** (*handshake*).



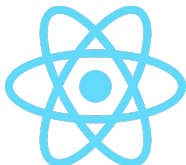


Tecnologías Frontend

- Robustez
- Modernidad



TypeScript



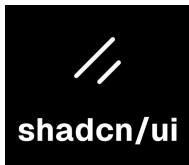
React



Vite



Vitest



shadcn/ui



React Flow



Lucide

Infraestructura y Colaboración

- Eficiencia
- Escalabilidad



Nginx



Node.js



WebRTC

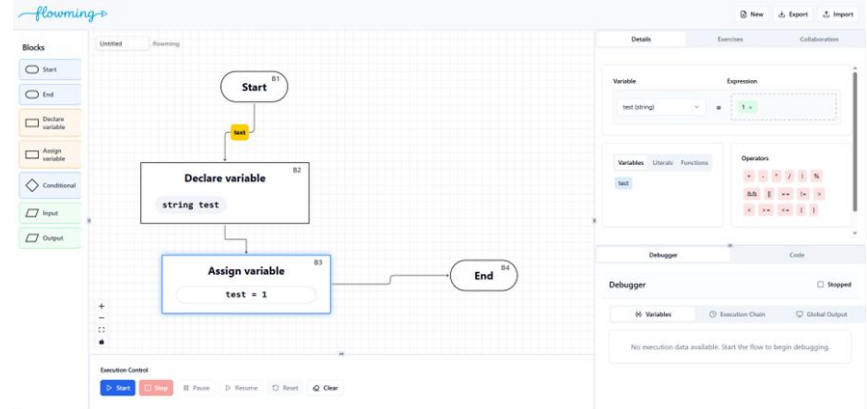


Yjs

Solución



Diseño de la solución

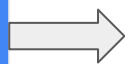


Versión inicial

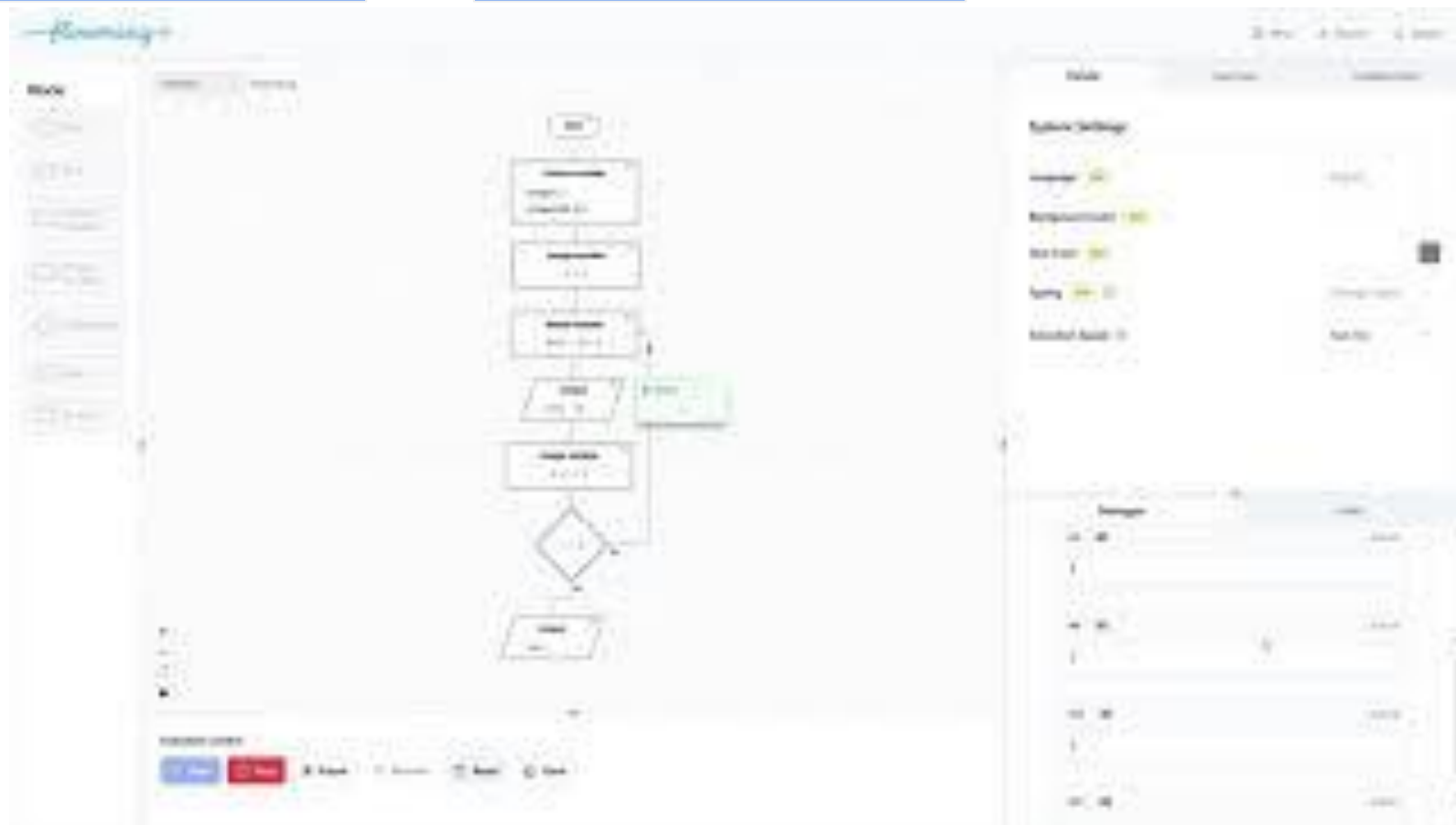
Versión “final”

3

Solución



Demostración



4

Validación

Evaluación con usuarios

TAREAS

1. Replicar diagrama dado
2. Crear algoritmo de cero
3. Ejecutar, depurar y probar el código

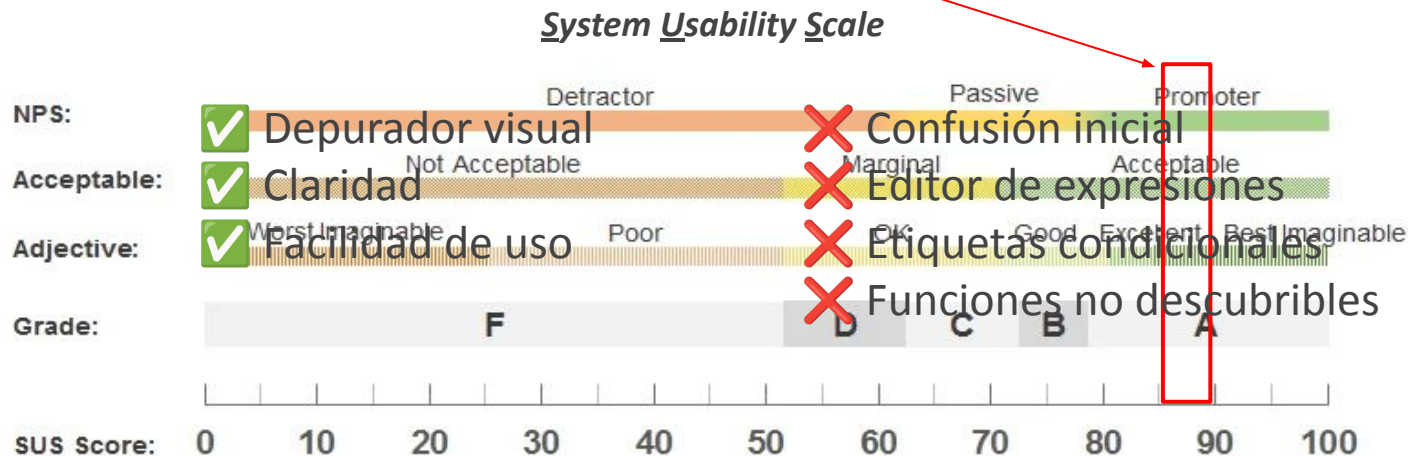
9 participantes

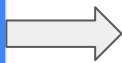
88,61 / 100

Puntuación Media SUS

PREGUNTAS ABIERTAS

- Más fácil o gustado
- Más costoso o confuso
- Sugerencias
- Errores





Pruebas Unitarias

- *Parser* (analizador) de **expresiones**
- Generador de **código** (Python)
- *Arrays* (listas)
- Modelos



Integración Continua (CI)

- Pruebas ejecutadas en cada **commit** y **pull request** (Github Actions)
- Previene la introducción de **errores** en la rama principal



Despliegue Continuo (CD)

- Despliegue automático de la versión **validada** en el servidor
- **Self-hosted runner** (VPN)
- Ciclo de entrega **rápido** y **fiable**



Objetivos Cumplidos

Herramienta **web funcional** que facilita el aprendizaje separando lógica y sintaxis.



Puente Efectivo a la Programación Textual

Generación de **código** Python equivalente al diagrama.



Validación Sólida

Notable **usabilidad** (SUS 88,61) y **robustez** técnica (pruebas y CI/CD)



Base para el Futuro

Base técnica **sólida** que permite futuras **expansiones** hacia una plataforma educativa completa.



Uso en futuros cursos de Programación del Grado en Ingeniería Mecánica en 2025/2026

5 Conclusiones

Líneas futuras



Mejoras de Usabilidad (corto plazo)

- Rediseñar **experiencia inicial**
- Corregir etiquetas de las **ramas condicionales**
- Diseño **totalmente responsive**
- Controles de depuración granulares
- Auditoría completa de **accesibilidad** (WCAG)



Ampliación Funcional (medio plazo)

- Soporte para **funciones** y recursividad
- Generación de código a **otros lenguajes** (Java, C++, etc.)



Evolución a Plataforma (largo plazo)

- Integración con plataformas **LMS** (e.g. Moodle)
- **Personalización** de la interfaz
- Sistema de **tipado débil** opcional



flowming →



¡Muchas gracias por su atención!



¿Preguntas?

Daniel Pérez Fernández [Redacted]

Repositorio: <https://github.com/MrPoll0/flowming>

Web: <https://dei.inf.uc3m.es/flowming/>