



---

## **TECNOLÓGICO NACIONAL DE MÉXICO**

INSTITUTO TECNOLÓGICO DE APIZACO

INTERFACES GRÁFICAS RESPONSIVAS PARA EL MÓDULO DE EVALUACIÓN  
DEL TEMA DE RESOLUCIÓN DE ECUACIONES LINEALES DE UN TUTOR  
COGNITIVO PARA ÁLGEBRA

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

**PARA OBTENER EL TÍTULO PROFESIONAL DE:**  
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y  
COMUNICACIONES

**PRESENTA:**  
ADOLFO PÉREZ PÉREZ

**NOMBRE DEL ASESOR:**  
DRA. BLANCA ESTELA PEDROZA MÉNDEZ



APIZACO, TLAX., MARZO 2026



Apizaco, Tlaxcala, 09/febrero/2026

Oficio No. S.C./OPV/0008/26

ASUNTO: Liberación de proyecto  
para titulación integral

**CECILIA YARELLI ARREGÍN LARA**  
JEFA DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES  
**P R E S E N T E.**

POR ESTE MEDIO LE INFORMO QUE HA SIDO LIBERADO EL SIGUIENTE PROYECTO PARA LA TITULACIÓN INTEGRAL.

|                             |                                                                                                                                               |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NOMBRE DEL EGRESADO:</b> | PÉREZ PÉREZ ADOLFO                                                                                                                            |
| <b>CARRERA:</b>             | INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES                                                                                  |
| <b>NO. DE CONTROL:</b>      | 20370998                                                                                                                                      |
| <b>NOMBRE DEL PROYECTO:</b> | Interfaces gráficas responsivas para el módulo de evaluación del tema de resolución de ecuaciones lineales de un tutor cognitivo para álgebra |
| <b>PRODUCTO U OPCIÓN:</b>   | TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL                                                                                         |

AGRADEZCO DE ANTEMANO SU VALIOSO APOYO EN ESTA IMPORTANTE ACTIVIDAD PARA LA FORMACIÓN PROFESIONAL DE NUESTROS EGRESADOS.

**ATENTAMENTE**

*Excelencia en Educación Tecnológica.*  
*Pensar para Servir, Servir para Triunfar®*

**MIQUELINA SÁNCHEZ PULIDO**  
JEFA DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

ccp. Coordinación de carrera  
Expediente  
MSP/mrol\*



SECRETARÍA DE EDUCACIÓN PÚBLICA  
TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE APIZACO  
SISTEMAS Y COMPUTACIÓN

**BLANCA ESTELA PEDROZA MÉNDEZ**  
ASESORA INTERNA



**2026**  
año de  
**Margarita**  
**Maza**





Tzompantepec, Tlaxcala, **17/Marzo/2026**

**ADOLFO PÉREZ PÉREZ**  
**EGRESADO(A) DE LA CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES**  
**CON NÚMERO DE CONTROL 20370998.**  
**P R E S E N T E.**

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** y cuyo tema es: **"INTERFACES GRÁFICAS RESPONSIVAS PARA EL MÓDULO DE EVALUACIÓN DEL TEMA DE RESOLUCIÓN DE ECUACIONES LINEALES DE UN TUTOR COGNITIVO PARA ÁLGEBRA"**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga la:

#### **AUTORIZACIÓN DE IMPRESIÓN**

Debiendo entregar **1 USB** del mismo, apropiadamente empastados, a la brevedad, para presentar su Acto de Recepción Profesional.

Sin otro particular por el momento me despido.

**ATENTAMENTE**

*Excelencia en Educación Tecnológica®*

*Pensar para Servir, Servir para Triunfar®*

**CECILIA YARELY ARREGUÍN LARA**

**JEFA DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES**



ccp. Archivo  
CYAL/Gvm\*



**2026**  
año de  
**Margarita**  
**Maza**





## Agradecimientos

Quiero comenzar expresando mi más profunda gratitud a Dios, por ser mi guía, mi fuerza y la fuente de bendiciones que hizo posible cada paso de este camino. A él le debo la oportunidad, la perseverancia y cada lección aprendida en el proceso.

Mi primer y más sentido reconocimiento es para la Dra. Blanca Estela Pedroza Méndez, asesora interna quien me recibió con confianza y me ofreció un espacio para aprender y crecer. Su apertura fue el primer paso de una experiencia invaluable. Al Dr. Juan Ramos Ramos, asesor externo y jefe del departamento, le agradezco cada observación y recomendación. Y no podría olvidar a cada integrante de este departamento, que, con sonrisas y consejos, me hicieron sentir parte de un equipo.

Agradezco al Lic. Alejandro Ruiz Castillo, por permitirme desarrollar mi experiencia laboral mientras cursaba la carrera y por su constante motivación.

A mi madre, Delia Pérez Ramírez, por ser mi fortaleza y nunca dejar de creer en mí; a mis tíos, el Ing. Antonio Pérez González y América Pérez González, por su respaldo inquebrantable; a mis abuelos, por su cariño y sus enseñanzas; y, por supuesto, a toda mi familia, porque en conjunto han sido mi red de apoyo más grande.

Por último, pero con un respeto enorme, mi gratitud se dirige al Instituto Tecnológico de Apizaco: estas aulas, estos laboratorios, estos espacios y —sobre todo— estos maestros y esta comunidad estudiantil me formaron. Aquí no solo adquirí herramientas técnicas y conocimientos académicos; aquí aprendí a pensar, a cuestionar, a soñar con proyectos y a tener la disciplina para ejecutarlos. Gracias por ser el terreno fértil donde pude sembrar mis esfuerzos y verlos germinar.

A todos, de corazón... ¡Gracias!



## Índice

|                                                                                       |           |
|---------------------------------------------------------------------------------------|-----------|
| Agradecimientos .....                                                                 | 2         |
| <b>CAPITULO I GENERALIDADES DEL PROYECTO .....</b>                                    | <b>5</b>  |
| 1. Resumen.....                                                                       | 6         |
| 2. Introducción.....                                                                  | 6         |
| 3. Descripción de la organización y del puesto o área de trabajo del estudiante ..... | 7         |
| 3.1 Descripción de la organización receptora.....                                     | 7         |
| 3.2 Descripción del área de trabajo y puesto desempeñado .....                        | 8         |
| 4. Problemas a resolver .....                                                         | 8         |
| 4.1 Problema principal .....                                                          | 9         |
| 4.2 Prioridad 1: Diseño e implementación de interfaces gráficas responsivas.....      | 9         |
| 4.3 Prioridad 2: Validación de responsividad y usabilidad en entornos diversos .....  | 9         |
| 4.4 Prioridad 3: Adaptación del algoritmo generador de ecuaciones .....               | 10        |
| 4.5 Prioridad 4: Integración del algoritmo en la interfaz gráfica.....                | 10        |
| 4.6 Prioridad 5: Documentación y entrega del módulo funcional .....                   | 10        |
| 5. Objetivos .....                                                                    | 11        |
| 5.1 Objetivo general .....                                                            | 11        |
| 5.2 Objetivos específicos .....                                                       | 11        |
| 6. Justificación.....                                                                 | 12        |
| <b>CAPITULO II MARCO TEÓRICO .....</b>                                                | <b>12</b> |
| 7. Marco teórico .....                                                                | 12        |
| 7.1 Fundamentos pedagógicos y educativos .....                                        | 12        |
| 7.2 Ecuaciones lineales con una variable .....                                        | 13        |
| 7.3 Diseño de interfaces gráficas de usuario (GUI) .....                              | 13        |
| 7.4 Desarrollo de software en Java para aplicaciones educativas.....                  | 14        |
| <b>CAPITULO III DESARROLLO E IMPLEMENTACIÓN .....</b>                                 | <b>14</b> |
| 8. Metodología .....                                                                  | 14        |
| 8.1 Fases del desarrollo.....                                                         | 15        |
| 8.2 Herramientas y técnicas .....                                                     | 16        |
| 8.3 Justificación de la metodología seleccionada.....                                 | 16        |
| 9. Configuración del entorno de desarrollo.....                                       | 17        |



|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| 9.1 Java Development Kit (JDK) 8/21 .....                                           | 18 |
| 9.2 NetBeans IDE 27 .....                                                           | 19 |
| 9.3 Canva (versión web) .....                                                       | 21 |
| 10. Entorno de pruebas embebidas (Raspberry Pi 4).....                              | 22 |
| 10.1 Raspberry Pi 4 Model B: .....                                                  | 23 |
| 10.2 Instalación de Raspberry Pi OS (64-bit): .....                                 | 23 |
| 10.3 Instalación de Java en Raspberry Pi: .....                                     | 26 |
| 10.4 Herramientas utilizadas para acceso remoto .....                               | 27 |
| 11. Configuración de acceso remoto .....                                            | 29 |
| 11.1 Dirección IP: Ejecución de hostname -I en terminal de Raspberry Pi. ....       | 29 |
| 11.2 Conexión con PuTTY .....                                                       | 30 |
| 11.3 Conexión con WinSCP.....                                                       | 31 |
| 11.4 Integración entre entornos.....                                                | 32 |
| 12. Diseño y Prototipado de interfaces .....                                        | 33 |
| 12.1 Proceso de diseño .....                                                        | 33 |
| 12.2 Principios de diseño aplicados .....                                           | 38 |
| 12.3 Salidas del proceso de prototipado .....                                       | 38 |
| 13. Programación de las interfaces gráficas .....                                   | 39 |
| 13.1 Preparación y procesamiento de recursos gráficos .....                         | 39 |
| 13.2 Patrón de desarrollo .....                                                     | 41 |
| 13.3 Sistema de animaciones .....                                                   | 41 |
| 13.4 Responsividad avanzada basada en porcentajes .....                             | 44 |
| 13.5 Gestión de capas visuales (orden Z) .....                                      | 45 |
| 13.6 Estructura de recursos específica para el tema 4 .....                         | 45 |
| 14. Programación del algoritmo para la generación de ecuaciones y su solución ..... | 46 |
| 14.1 Análisis del algoritmo base proporcionado .....                                | 46 |
| 14.2 Proceso de solución al algoritmo.....                                          | 47 |
| 14.3 Proceso de adaptación y simplificación.....                                    | 48 |
| 15. Integración del algoritmo en las interfaces .....                               | 51 |
| 15.1 Punto de integración principal: int1Ret1T4SiEEAI.....                          | 51 |
| 16. Pruebas de funcionamiento de las interfaces .....                               | 57 |
| 16.1 Estrategia de pruebas implementada .....                                       | 57 |



|                                                                |           |
|----------------------------------------------------------------|-----------|
| <b>16.2 Pruebas del algoritmo generador de ecuaciones.....</b> | <b>57</b> |
| <b>16.3 Pruebas de la interfaz gráfica .....</b>               | <b>58</b> |
| <b>16.4 Pruebas de manejo de errores.....</b>                  | <b>60</b> |
| <b>16.5 Pruebas en Raspberry Pi 4.....</b>                     | <b>61</b> |
| <b>17. Pruebas de usabilidad .....</b>                         | <b>65</b> |
| <b>17.1 Metodología de pruebas de usabilidad.....</b>          | <b>65</b> |
| <b>17.2 Escenarios de prueba .....</b>                         | <b>65</b> |
| <b>17.3 Criterios de usabilidad evaluados.....</b>             | <b>66</b> |
| <b>CONCLUSIONES.....</b>                                       | <b>67</b> |
| <b>Conclusiones .....</b>                                      | <b>67</b> |
| <b>Recomendaciones.....</b>                                    | <b>68</b> |
| <b>Experiencia adquirida .....</b>                             | <b>69</b> |
| <b>Competencias desarrolladas .....</b>                        | <b>70</b> |
| <b>Referencias .....</b>                                       | <b>71</b> |

# **CAPITULO I GENERALIDADES DEL PROYECTO**

## **1. Resumen**

Este reporte presenta los resultados de las residencias profesionales realizadas en el Departamento de Posgrado e Investigación del Instituto Tecnológico de Apizaco, enfocadas en el desarrollo e implementación de “Interfaces gráficas responsivas para el módulo de evaluación del tema de resolución de ecuaciones lineales de un tutor cognitivo para álgebra”. El objetivo principal fue diseñar, programar e integrar interfaces gráficas responsivas para la generación, visualización y evaluación automatizada de ecuaciones lineales con una incógnita.

Las actividades incluyeron: 1) Prototipado de interfaces para el tema “Problemas con una variable”, 2) Programación de las Interfaces en Java con NetBeans IDE; 3) Análisis, adaptación e implementación de un algoritmo que genera ecuaciones aleatorias 4) Integración del módulo en la arquitectura del sistema existente; 5) Pruebas de usabilidad y funcionalidad. Para garantizar la responsividad multiplataforma, se configuró un entorno de pruebas en Raspberry Pi 4 con Raspberry Pi OS, realizando instalación del JDK, ejecución de scripts, validación gráfica y transferencia de archivos mediante WinSCP/SSH.

Como resultado, se entregó un módulo funcional, probado y completamente integrado, que permite a los estudiantes practicar la resolución de ecuaciones lineales mediante una interfaz intuitiva y retroalimentación inmediata. La contribución fortaleció el sistema educativo existente y demostró competencias en desarrollo de software modular, integración de algoritmos y validación en entornos embebidos.

## 2. Introducción

Las residencias profesionales representan un componente esencial en la formación integral del ingeniero, al constituir el puente entre los conocimientos teóricos adquiridos en las aulas y su aplicación en entornos laborales y de investigación reales. Este periodo no solo fortalece las competencias técnicas y profesionales del estudiante, sino que también contribuye al desarrollo de proyectos concretos dentro de las instituciones receptoras.

El presente documento tiene como finalidad dar cuenta de las actividades realizadas durante el periodo de residencias profesionales en el Departamento de Posgrado e Investigación del Instituto Tecnológico de Apizaco, llevadas a cabo bajo la supervisión de la Dra. Blanca Estela Pedroza Méndez (asesora interna) y el Dr. Juan Ramos Ramos (asesor externo). El trabajo se centró en el desarrollo e implementación del módulo "Problemas con una variable", como parte de un sistema educativo preexistente diseñado para la enseñanza y práctica del álgebra.

El proyecto surgió de la necesidad de ampliar dicho sistema, el cual ya contaba con varios temas implementados, pero carecía de un módulo dedicado específicamente a la resolución de ecuaciones lineales de primer grado. Para ello, fue necesario diseñar interfaces gráficas amigables y responsivas, adaptar un algoritmo generador de ecuaciones, integrarlo al sistema y validar su funcionamiento en diferentes plataformas, incluyendo equipos de escritorio y dispositivos embebidos como la Raspberry Pi 4.

A lo largo de este reporte se describen de manera estructurada los aspectos generales del proyecto, la organización receptora, los problemas abordados, los objetivos planteados, la justificación del trabajo y, posteriormente, el desarrollo técnico, los resultados obtenidos y las conclusiones derivadas de esta experiencia. La realización de estas residencias no solo permitió aplicar conocimientos en programación, diseño de interfaces y gestión de proyectos, sino que también representó una valiosa oportunidad para contribuir al fortalecimiento de las herramientas educativas del Departamento.

### **3. Descripción de la organización y del puesto o área de trabajo del estudiante**

#### **3.1 Descripción de la organización receptora**

Las residencias profesionales se llevaron a cabo en el Instituto Tecnológico de Apizaco (ITA), institución pública de educación superior perteneciente al Tecnológico Nacional de México., que se ha consolidado como un referente en la formación de ingenieros y profesionales tecnológicos en la región, con una oferta académica que incluye ingenierías, licenciaturas y posgrados en áreas como sistemas computacionales, industrial, electrónica, mecatrónica y gestión empresarial.

##### **Misión:**

“Formar profesionistas que sean líderes visionarios, emprendedores, con capacidad de generar soluciones disruptivas, innovando en la ciencia y la tecnología, comprometidos con el desarrollo económico, responsabilidad social teniendo una visión sostenible y sustentable.”

##### **Visión:**

“Ser una Institución de educación superior tecnológica de vanguardia, formadora de ciudadanos del mundo, con conocimientos intercultural, respetando la diversidad social.”

#### **3.2 Descripción del área de trabajo y puesto desempeñado**

El estudiante se integró al equipo del Departamento de Posgrado e Investigación, bajo la supervisión directa de la Dra. Blanca Estela Pedroza Méndez (asesora interna) y la coordinación general del Dr. Juan Ramos Ramos, jefe del departamento.

##### **Puesto desempeñado**

Residente de Desarrollo de Software.

## 4. Problemas a resolver

Durante el análisis inicial del proyecto y con base a la descripción en el anteproyecto proporcionado por el Departamento:

“Actualmente con el manejo de diversos dispositivos electrónicos la interacción con los usuarios requiere interfaces responsivas es decir interfaces que se adapten al tamaño del dispositivo de cómputo que se esté utilizando, además de aplicar muchas de las técnicas de la interacción humana máquina para el diseño de interfaces gráficas el hacer uso de interfaces responsivas permite que dichas interfaces tengan una mejor usabilidad así mismo dentro del contexto de tutoriales cognitivos además de integrar El paradigma de interfaces responsivas también es importante integrar algunas características que permitan evaluar la adquisición del conocimiento en el usuario por lo tanto el objetivo de este proyecto es diseñar interfaces que de manera automática y obligatoria generen diferentes ejercicios relacionados con el tema de la solución de ecuaciones lineales aplicando algunas técnicas de despeje, es evidente que para que el sistema pueda evaluar las respuestas del usuario son correctas el sistema debe tener implementadas las estrategias para que el mismo tiempo que genere la ecuación aleatoria y genere la solución y de esta manera evaluar si el usuario está haciendo correctamente el proceso de solución para la implementación de las interfaces se puede hacer uso de alguna herramienta que permite el diseño animado y vistoso de interfaces gráficas de usuario como puede ser NetBeans con Java o algún otro lenguaje”

Se identificaron los siguientes problemas técnicos y funcionales que requirieron solución:

### 4.1 Problema principal

El sistema ya contaba con temas implementados, pero carecía de un apartado dedicado específicamente a la práctica y evaluación de ecuaciones con una variable, limitando su utilidad para estudiantes que se inician en el álgebra elemental.

### 4.2 Prioridad 1: Diseño e implementación de interfaces gráficas responsivas

- Problema:** Según lo establecido en el anteproyecto, las interfaces debían adaptarse a diferentes dispositivos de cómputo y cumplir con principios de interacción humano-máquina para garantizar usabilidad.

**•Solución requerida:**

Prototipar interfaces atractivas y funcionales en una herramienta de diseño (Canva).

Programar las interfaces en Java (NetBeans IDE) asegurando que sean responsivas.

Integrar imágenes y componentes visuales optimizados para diferentes resoluciones.

**4.3 Prioridad 2: Validación de responsividad y usabilidad en entornos diversos**

**•Problema:** El sistema debía probarse no solo en equipos de escritorio, sino también en dispositivos con recursos limitados y diferentes tamaños de pantalla, asegurando que la experiencia fuera consistente.

**•Solución requerida:**

a) Configurar un entorno de pruebas en una Raspberry Pi 4 con Raspberry Pi OS.

b) Realizar pruebas de usabilidad en al menos tres resoluciones distintas.

c) Validar la ejecución del archivo .jar en diferentes plataformas.

**4.4 Prioridad 3: Adaptación del algoritmo generador de ecuaciones**

- **Problema:** El algoritmo existente generaba ecuaciones aleatorias pero incluía exponentes y operaciones propias de polinomios, no alineadas con ecuaciones de primer grado.
- **Solución requerida:** Modificar el algoritmo para que genere exclusivamente ecuaciones lineales de la forma  $AX + B = C$ , con coeficientes enteros o racionales, y que además genere automáticamente la solución paso a paso.

#### 4.5 Prioridad 4: Integración del algoritmo en la interfaz gráfica

•**Problema:** El algoritmo adaptado debía conectarse de manera fluida con la capa gráfica, permitiendo la generación aleatoria de ejercicios, la captura de respuestas del usuario y la evaluación automática.

•**Solución requerida:** Diseñar una arquitectura de comunicación entre el front-end (interfaz) y el back-end (lógica del algoritmo), garantizando que cada ecuación generada tuviera su solución precalculada para validar la respuesta del usuario.

#### 4.6 Prioridad 5: Documentación y entrega del módulo funcional

•**Problema:** El módulo debía entregarse completamente integrado al sistema educativo, con documentación clara para futuros mantenimientos o mejoras.

•**Solución requerida:** Elaborar la documentación técnica del código y reporte de pruebas.

Los problemas priorizados anteriormente derivan directamente de este objetivo, asegurando que el módulo final cumpla con:

- Generación automática y aleatoria de ecuaciones.
- Solución precalculada.
- Interfaces responsivas y usables.
- Integración completa en el sistema.

## **5. Objetivos**

### **5.1 Objetivo general**

Implementar las interfaces gráficas responsivas para evaluar el tema de resolución de ecuaciones lineales en algún lenguaje de programación.

### **5.2 Objetivos específicos**

- Realizar el prototipado de las interfaces.
- Programar las interfaces.
- Analizar el algoritmo para la generación de las ecuaciones y su solución.
- Programar el algoritmo y ligarlo con las interfaces diseñadas.
- Realizar pruebas de usabilidad de las interfaces
- Escribir resultados

## **6. Justificación**

El proceso de evaluación de ejercicios relacionados con temas de matemáticas no siempre es fácil para el docente. Por lo tanto, el diseño de una herramienta que permita hacer este proceso, de manera automática, permite que el docente pueda tener evidencias, un poco más certeras del avance del alumno, ya que el generar ejercicios aleatorios, evita que los usuarios puedan compartir las respuestas, y sea un poco más eficiente.

En síntesis, este proyecto no solo responde a una necesidad concreta dentro del proceso de enseñanza-aprendizaje del álgebra, sino que también constituyó una oportunidad de aplicación práctica de conocimientos en programación, diseño de interfaces, algoritmos y gestión de proyectos software, alineados con la formación integral del alumno.

## CAPITULO II MARCO TEÓRICO

### 7. Marco teórico

#### 7.1 Fundamentos pedagógicos y educativos

##### Enseñanza de las matemáticas mediante tecnología educativa

La integración de herramientas digitales en la enseñanza de las matemáticas ha demostrado mejorar la comprensión conceptual, la motivación y el aprendizaje autónomo (Brendan, 2016). Los tutoriales cognitivos como el sistema en el que se integró este módulo combinan explicaciones guiadas con práctica interactiva, permitiendo una retroalimentación inmediata que facilita la corrección de errores y la consolidación de conocimientos.

##### Evaluación automatizada en el aprendizaje de álgebra

La evaluación automatizada de ejercicios matemáticos permite no solo liberar tiempo docente, sino también ofrecer a los estudiantes múltiples oportunidades de práctica con variabilidad en los problemas. La generación aleatoria de ejercicios evita la memorización de respuestas y fomenta la comprensión profunda de los procedimientos (J., 2013).

#### 7.2 Ecuaciones lineales con una variable

##### Definición

Una ecuación lineal con una variable puede expresarse en la forma general (Cuemath.com, 2025)

$$AX + B = C$$

donde  $a$ ,  $b$  y  $c$  son constantes ( $a \neq 0$ ) y  $x$  es la incógnita. La solución se obtiene mediante operaciones algebraicas de despeje:  $X = C - B/A$

## 7.3 Diseño de interfaces gráficas de usuario (GUI)

### Principios de Interacción Humano-Computadora (IHC)

El diseño de interfaces efectivas se rige por principios como la usabilidad, la consistencia, la retroalimentación visual y la minimización de la carga cognitiva (Donald, 2013). En contextos educativos, es crucial que la interfaz sea intuitiva para que el usuario se centre en el contenido matemático, no en la operación del software.

### Interfaces gráficas responsivas (Responsive GUI)

Una interfaz responsiva se adapta dinámicamente al tamaño y resolución de la pantalla del dispositivo, manteniendo la funcionalidad y legibilidad (Uxcel, 2025). Esto se logra mediante:

- Layouts flexibles
- Componentes escalables

## 7.4 Desarrollo de software en Java para aplicaciones educativas

### Java

Java proporciona las bibliotecas AWT (Abstract Window Toolkit) y Swing para el desarrollo de interfaces gráficas de escritorio. Swing, construido sobre AWT, ofrece componentes más modernos y personalizables, además de una arquitectura *pluggable look-and-feel* que facilita la adaptación visual (MSMK, 2025).

### NetBeans IDE como entorno de desarrollo

NetBeans incluye un diseñador visual de GUI (GUI Builder) que acelera la creación de interfaces mediante arrastrar y soltar, generando automáticamente código Java. Esto permite concentrarse en la lógica de la aplicación y en la experiencia de usuario (NetBeans, Introduction to GUI Building, 2025).

# CAPITULO III DESARROLLO E IMPLEMENTACIÓN

## 8. Metodología

(Valtx, 2022) y (Jiménez-Hernández, 2012). Para el desarrollo del módulo "Problemas con una variable" se adoptó una metodología híbrida que combinó elementos de:

- Scrum (para la gestión iterativa del proyecto y reuniones de seguimiento).
- Desarrollo iterativo e incremental (por la naturaleza modular del software).
- Prototipado rápido (dado el componente de diseño de interfaces).

Este enfoque permitió adaptarse a requerimientos cambiantes, mantener una comunicación fluida con los asesores y entregar valor de manera progresiva.

### 8.1 Fases del desarrollo

#### Fase 1: Análisis y planificación

- Revisión del anteproyecto y requisitos del módulo.
- Establecimiento de objetivos y cronograma en coordinación con los asesores.
- Definición de las herramientas tecnológicas (Java, NetBeans, Canva, Raspberry Pi).

#### Fase 2: Diseño

- Creación de prototipos de interfaz en Canva.
- Validación de diseños con la asesora.
- Análisis del algoritmo y plan de adaptación.

#### Fase 3: Desarrollo iterativo

- **Sprint planning:** Definición de tareas para el ciclo.
- **Desarrollo técnico:** Programación de interfaces y algoritmo.

- **Revisiones diarias breves** (seguimiento personal con asesora).
- **Demo al final del sprint:** Presentación de avances a la asesora.

#### **Fase 4: Integración y pruebas**

- Integración del algoritmo con la interfaz gráfica.
- Pruebas de funcionalidad en entorno local.
- Configuración y pruebas en Raspberry Pi 4.
- Pruebas de usabilidad en diferentes dispositivos.

#### **Fase 5: Documentación y entrega.**

- Documentación técnica del código.
- Preparación del reporte final.

### **8.2 Herramientas y técnicas**

#### **Para la gestión del proyecto**

- **Diagrama de Gantt** (físico/digital) para seguimiento de tareas.
- **Reuniones semanales de seguimiento** con la Dra. Blanca Estela Pedroza Méndez.
- **Comunicación constante** vía mensajería.

#### **Para el desarrollo técnico**

- Java y NetBeans IDE para desarrollo las interfaces gráficas.
- **Canva** para prototipada visual.
- **Raspberry Pi 4** con Raspberry Pi OS para pruebas de responsividad.
-

### 8.3 Justificación de la metodología seleccionada

La combinación de elementos ágiles (Scrum) con desarrollo iterativo fue adecuada porque:

1. **Flexibilidad:** Permitía ajustar diseños y funcionalidades basándose en retroalimentación constante.
2. **Entrega incremental:** Cada sprint producía un entregable tangible (prototipos, interfaz funcional, algoritmo adaptado).
3. **Control de calidad:** Las pruebas continuas garantizaron un producto estable.
4. **Comunicación efectiva:** Las reuniones periódicas con los asesores aseguraron la alineación con los objetivos del departamento.

## 9. Configuración del entorno de desarrollo

Para el desarrollo del módulo "Problemas con una variable" fue necesario configurar dos entornos complementarios: un entorno principal de desarrollo en un sistema Windows y un entorno de pruebas embebidas en Raspberry Pi 4. Esta configuración dual permitió tanto la programación eficiente como la validación de responsividad en plataformas heterogéneas.

### Entorno principal de desarrollo (Windows 10/11)

El entorno principal se configuró en una computadora portátil con sistema operativo Windows, donde se realizó la mayor parte de la programación y pruebas iniciales.

### Herramientas instaladas y configuradas:

#### 9.1 Java Development Kit (JDK) 8/21

- Propósito: Proporcionar el compilador y las bibliotecas necesarias para desarrollar aplicaciones Java.
- Proceso de instalación:



*Ilustración 1 Java*

Descarga del instalador oficial desde Oracle (Oracle, 2025).

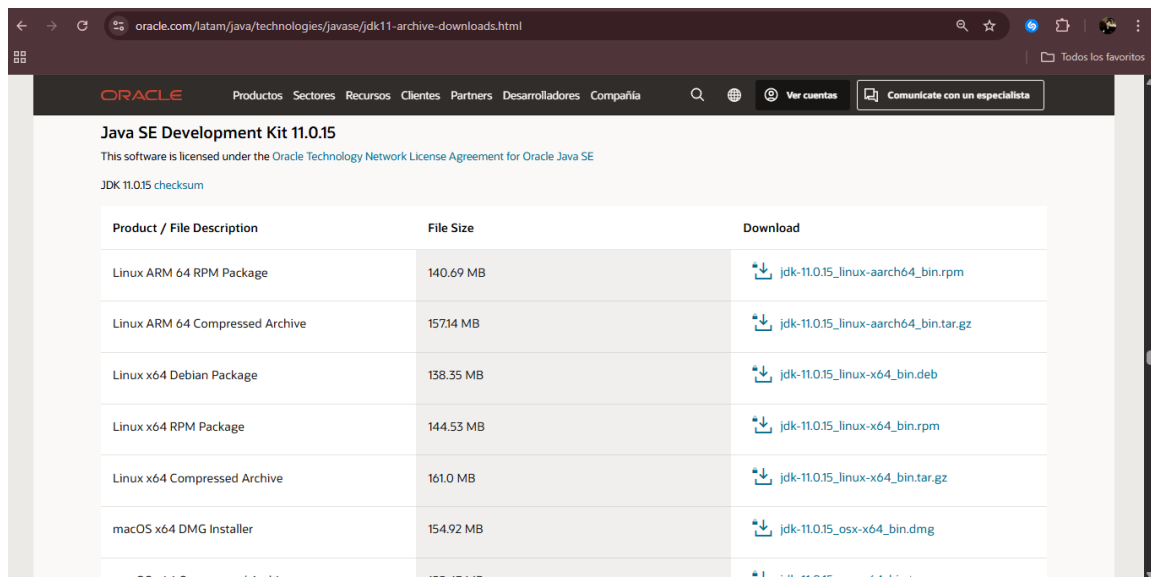


Ilustración 2 Sitio Web de Oracle

Ejecución del instalador con configuración personalizada y configuración de variables de entorno (JAVA\_HOME, PATH) para acceso desde línea de comandos.

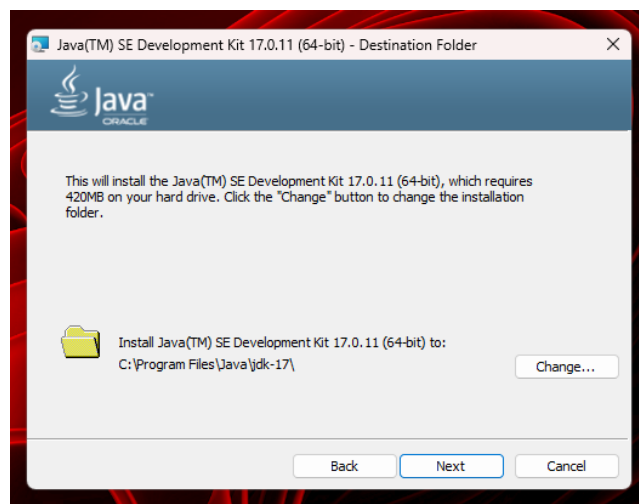


Ilustración 3 Instalación de JDK 17

Verificación: Ejecución de `java -version` y `javac -version`, para confirmar la instalación correcta.

```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Versión 10.0.26100.7171]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\adolf>java -version
java version "17.0.11" 2024-04-16 LTS
Java(TM) SE Runtime Environment (build 17.0.11+7-LTS-207)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.11+7-LTS-207, mixed mode, sharing)

C:\Users\adolf>
```

*Ilustración 4 Consulta de instalación en CMD*

## 9.2 NetBeans IDE 27

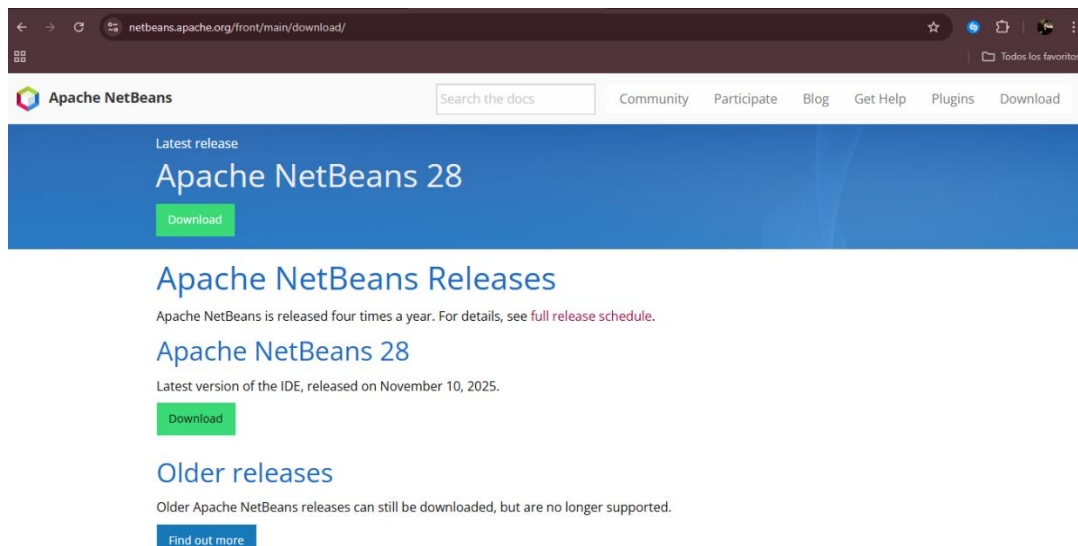
- **Propósito:** Entorno de desarrollo integrado con diseñador visual de GUI para acelerar la creación de interfaces.



*Ilustración 5 Apache NetBeans*

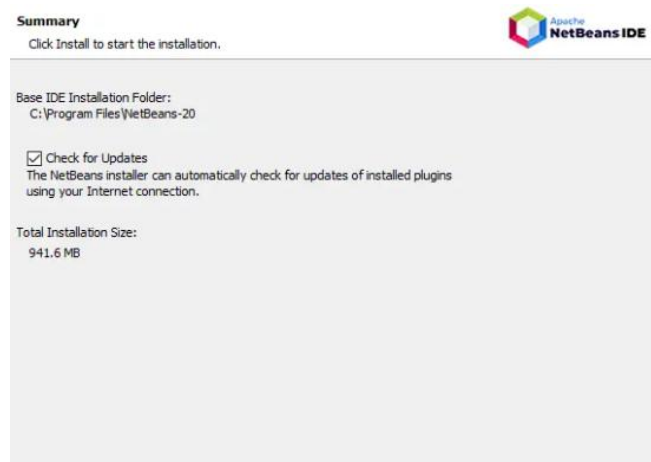
- **Proceso de instalación:**

Descarga del instalador completo desde el sitio oficial de Apache NetBeans (NetBeans, Apache NetBeans releases., 2025):



*Ilustración 6 Sitio Web Apache NetBeans*

- **Instalación con todos los módulos estándar para desarrollo Java SE.**  
Configuración del JDK previamente instalado como runtime por defecto.



*Ilustración 7 Instalación de NetBeans*

- **Configuración adicional:**
  - Ajuste de temas y tamaño de fuente para comodidad visual.
  - Configuración de atajos de teclado personalizados.

### 9.3 Canva (versión web)

- **Propósito:** Herramienta de diseño gráfico en línea para el prototipado rápido de interfaces.



Ilustración 8 Canva

- **Configuración:** Acceso mediante navegador web con cuenta institucional.

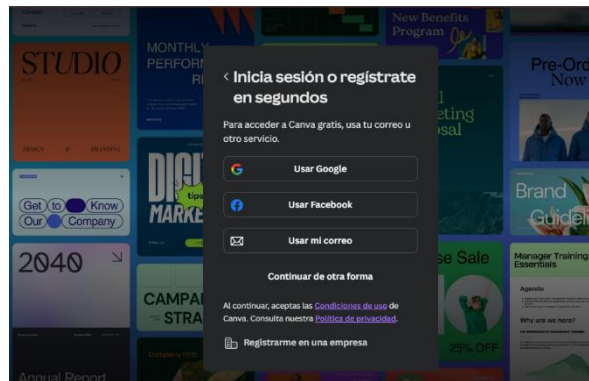


Ilustración 9 Canva Login

- **Uso específico:** Creación de layouts, selección de paleta de colores institucional y diseño de elementos visuales (Canva, 2025).

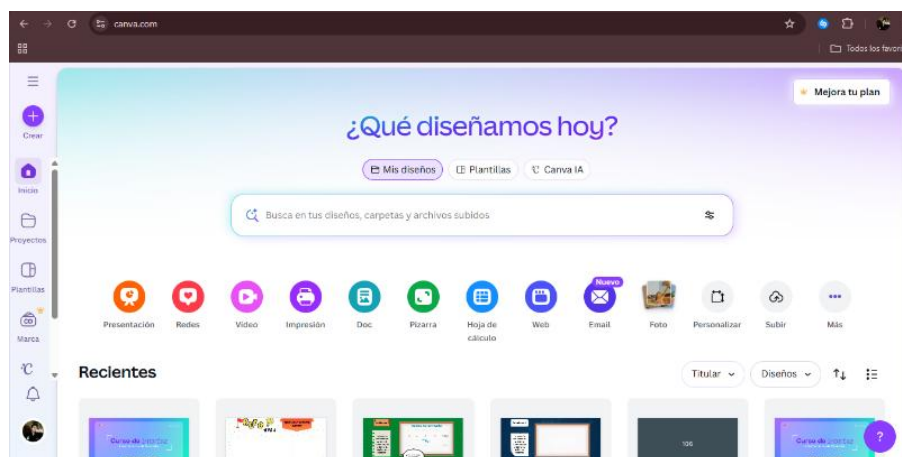


Ilustración 10 Canva Home

## 10. Entorno de pruebas embebidas (Raspberry Pi 4)

Para validar la responsividad y portabilidad del módulo, se configuró una Raspberry Pi 4 Model B con 4GB de RAM.

Ya que la Raspberry Pi 4 Model B es una computadora de placa reducida (*Single Board Computer – SBC*) desarrollada por la Fundación Raspberry Pi, su objetivo principal es proporcionar una plataforma accesible y versátil para el aprendizaje de la informática, la electrónica y el desarrollo de proyectos de bajo costo (Pi, ¿Que es Raspberry Pi?, 2019).



*Ilustración 11 Raspberry Pi*

### Configuración del hardware y software:

#### 10.1 Raspberry Pi 4 Model B:

- **Especificaciones:** 4GB RAM, almacenamiento microSD de 32GB.
- **Conexiones:** Monitor HDMI, teclado y mouse USB, alimentación mediante adaptador oficial.



*Ilustración 12 Raspberry Pi 4 Model B*

## 10.2 Instalación de Raspberry Pi OS (64-bit):

- **Proceso:**

- Descarga e instalar (en Windows) Raspberry Pi Imager desde su sitio web oficial de Raspberry (Pi, Downloads, 2025).

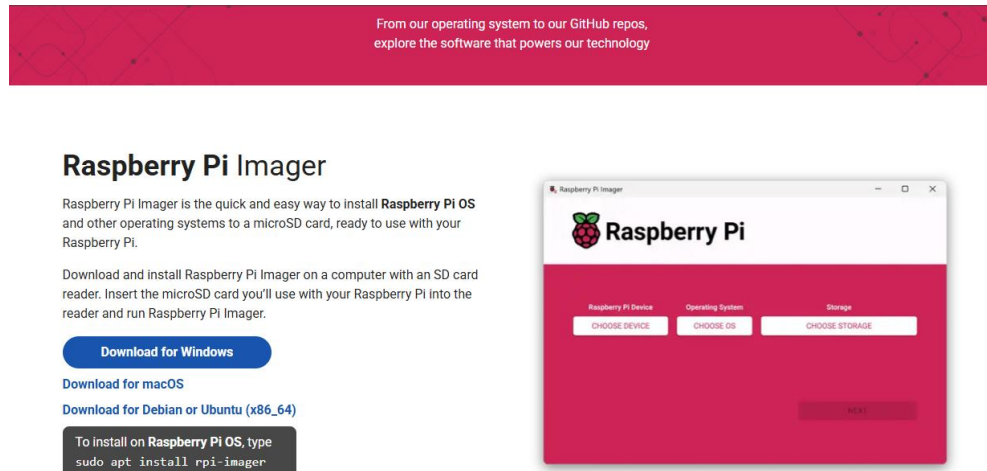


Ilustración 13 Sitio Web Raspberri Pi

- Grabación en microSD utilizando Raspberry Pi Imager en Windows. Seleccionar el sistema operativo **Raspberry Pi OS (64-bit)** en la opción recomendada y escoger la tarjeta microSD como destino.

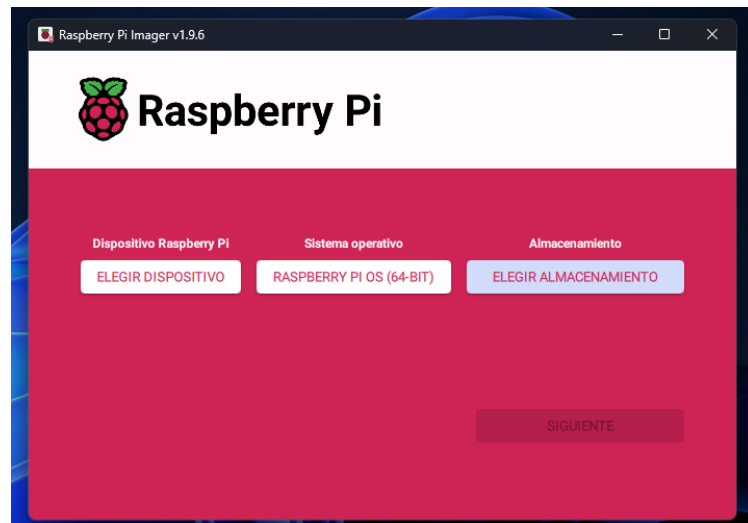
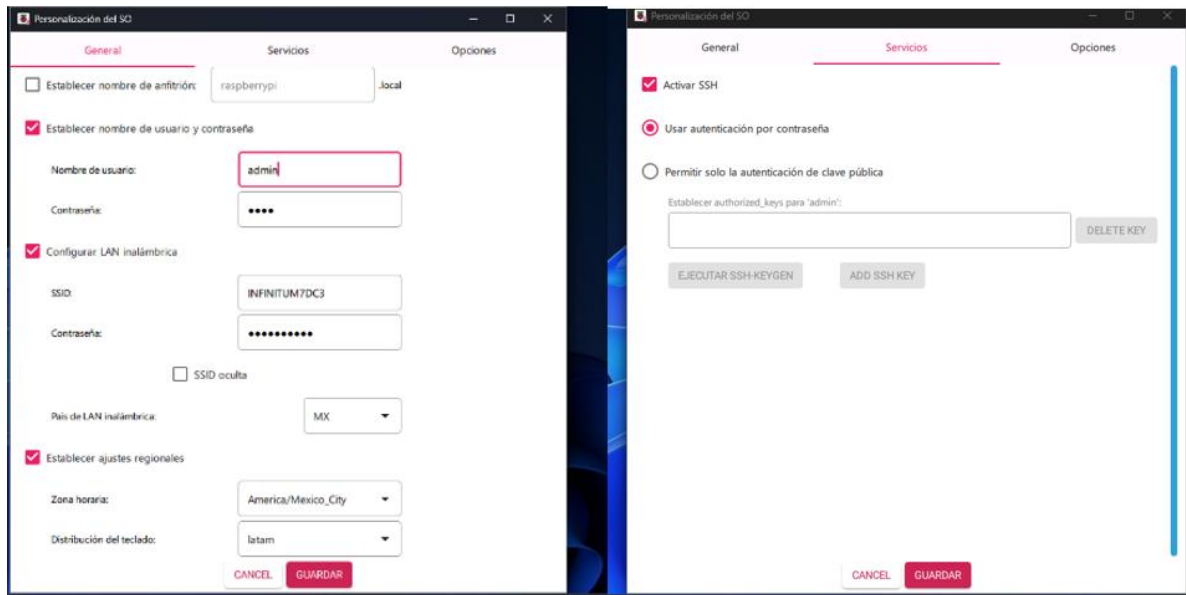


Ilustración 14 Raspberry Pi Imager

- Configuración inicial del sistema (idioma, zona horaria, usuario, contraseña). Muy importante Activar SSH en el dispositivo, ya que más adelante nos ayudara para compartir archivos de remota, lo cual será más fácil y rápido



*Ilustración 15 Configuración Raspberry Pi Imager*

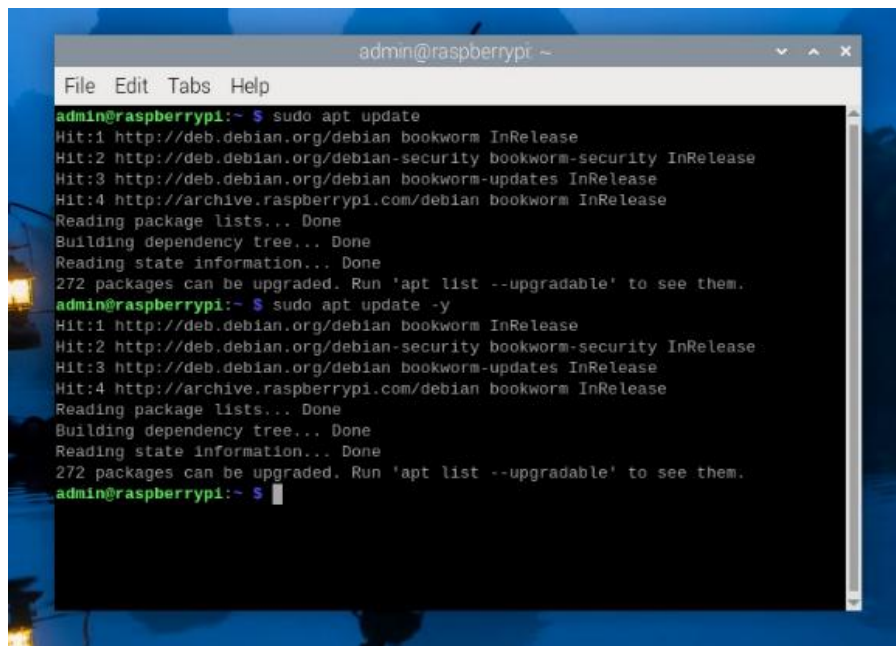
Al final podremos acceder al nuestro nuevo sistema operativo en la Raspberry Pi:



*Ilustración 16 Interfaz de Raspberry Pi OS*

- **Actualizaciones:** Es importante ejecutar los siguientes comandos en la consola de nuestro Raspberry Pi OS: `sudo apt update` & `sudo apt upgrade` para actualizar paquetes dentro del sistema.

```
sudo apt update  
sudo apt upgrade -y
```



```
admin@raspberrypi: ~  
File Edit Tabs Help  
admin@raspberrypi:~$ sudo apt update  
Hit:1 http://deb.debian.org/debian bookworm InRelease  
Hit:2 http://deb.debian.org/debian-security bookworm-security InRelease  
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease  
Hit:4 http://archive.raspberrypi.com/debian bookworm InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
272 packages can be upgraded. Run 'apt list --upgradable' to see them.  
admin@raspberrypi:~$ sudo apt upgrade -y  
Hit:1 http://deb.debian.org/debian bookworm InRelease  
Hit:2 http://deb.debian.org/debian-security bookworm-security InRelease  
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease  
Hit:4 http://archive.raspberrypi.com/debian bookworm InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
272 packages can be upgraded. Run 'apt list --upgradable' to see them.  
admin@raspberrypi:~$
```

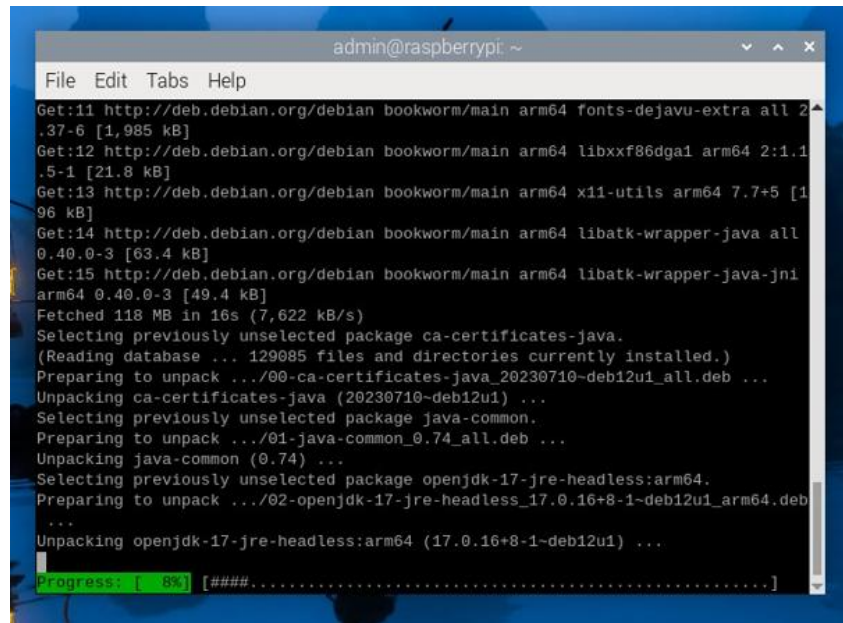
*Ilustración 17 Actualización de paquetes*

### 10.3 Instalación de Java en Raspberry Pi:

- **JDK 17 (OpenJDK):** Instalación mediante `sudo apt install default-jdk -y`

```
sudo apt install default-jdk -y
```

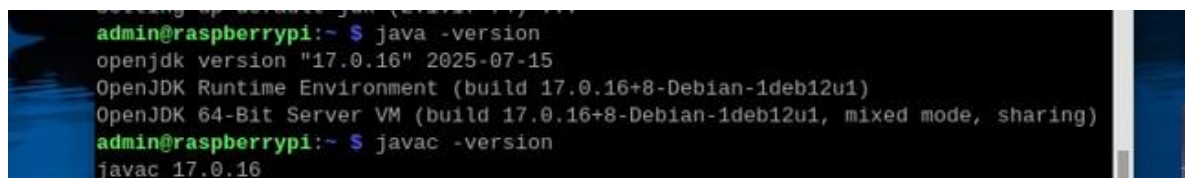
Este paquete instala automáticamente la versión más reciente de OpenJDK disponible en los repositorios de Raspberry Pi OS (al momento de la instalación, OpenJDK 17).



```
admin@raspberrypi: ~
File Edit Tabs Help
Get:11 http://deb.debian.org/debian bookworm/main arm64 fonts-dejavu-extra all 2
.37-6 [1,985 kB]
Get:12 http://deb.debian.org/debian bookworm/main arm64 libxxf86dga1 arm64 2:1.1
.5-1 [21.8 kB]
Get:13 http://deb.debian.org/debian bookworm/main arm64 x11-utils arm64 7.7+5 [1
96 kB]
Get:14 http://deb.debian.org/debian bookworm/main arm64 libatk-wrapper-java all
0.40.0-3 [63.4 kB]
Get:15 http://deb.debian.org/debian bookworm/main arm64 libatk-wrapper-java-jni
arm64 0.40.0-3 [49.4 kB]
Fetched 118 MB in 16s (7,622 kB/s)
Selecting previously unselected package ca-certificates-java.
(Reading database ... 129085 files and directories currently installed.)
Preparing to unpack .../00-ca-certificates-java_20230710-deb12u1_all.deb ...
Unpacking ca-certificates-java (20230710-deb12u1) ...
Selecting previously unselected package java-common.
Preparing to unpack .../01-java-common_0.74_all.deb ...
Unpacking java-common (0.74) ...
Selecting previously unselected package openjdk-17-jre-headless:arm64.
Preparing to unpack .../02-openjdk-17-jre-headless_17.0.16+8-1-deb12u1_arm64.deb
...
Unpacking openjdk-17-jre-headless:arm64 (17.0.16+8-1-deb12u1) ...
Progress: [ 8%] [####.....]
```

*Ilustración 18 Instalación JDK en Raspberri Pi*

- **Verificación:** java -version y javac -version en terminal.



```
admin@raspberrypi:~ $ java -version
openjdk version "17.0.16" 2025-07-15
OpenJDK Runtime Environment (build 17.0.16+8-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.16+8-Debian-1deb12u1, mixed mode, sharing)
admin@raspberrypi:~ $ javac -version
javac 17.0.16
```

*Ilustración 19 Consulta de versión Java*

## 10.4 Herramientas utilizadas para acceso remoto

- **PuTTY:** Para acceso mediante línea de comandos vía SSH.
  - **Acceder por SSH:** conectarse de forma segura a la terminal de la Raspberry desde otra computadora.
  - **Administrar equipos a distancia:** ejecutar comandos como si estuvieras físicamente frente al dispositivo.
  - **Ahorrar periféricos:** no necesitas monitor, teclado ni ratón para la Raspberry, ya que todo lo controlas desde tu PC.
  - **Probar conexiones:** también sirve para verificar que un servidor o dispositivo de red está respondiendo correctamente.

Descarga e instala desde su sitio oficial (PuTTY, 2025):

**Download PuTTY: latest release (0.83)**

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)  
Download: [Stable](#) - [Snapshot](#) | [Docs](#) | [Privacy](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.83, released on 2025-02-08.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [permanent link to the 0.83 release](#).

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

**Package files**

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download at the [Microsoft Store](#); they usually take a few days to appear there after we release them.

**MSI ("Windows Installer")**

|             |                                                |                             |
|-------------|------------------------------------------------|-----------------------------|
| 64-bit x86: | <a href="#">putty-64bit-0.83-installer.msi</a> | <a href="#">(signature)</a> |
| 64-bit Arm: | <a href="#">putty-arm64-0.83-installer.msi</a> | <a href="#">(signature)</a> |
| 32-bit x86: | <a href="#">putty-0.83-installer.msi</a>       | <a href="#">(signature)</a> |

**Unix source archive**

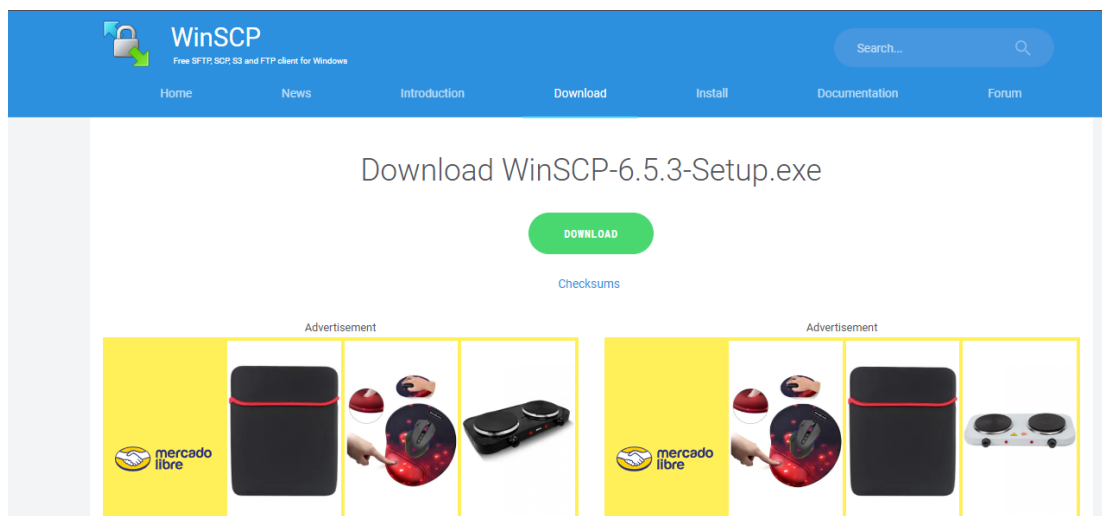
|          |                                   |                             |
|----------|-----------------------------------|-----------------------------|
| .tar.gz: | <a href="#">putty-0.83.tar.gz</a> | <a href="#">(signature)</a> |
|----------|-----------------------------------|-----------------------------|

Ilustración 20 Sitio Web PuTTY

- **WinSCP** Para transferencia segura de archivos entre Windows y Raspberry Pi mediante protocolo SCP/SFTP.

Gratuito para Windows que permite transferir archivos de forma segura entre una computadora con Windows y una Raspberry Pi u otro servidor remoto. Ofrece una interfaz gráfica intuitiva similar al Explorador de Windows, haciendo que la copia de archivos sea tan simple como arrastrar y soltar. Soporta protocolos seguros como SSH para transferencias cifradas.

Descargado e instalado desde su sitio web (WinSCP, 2025):



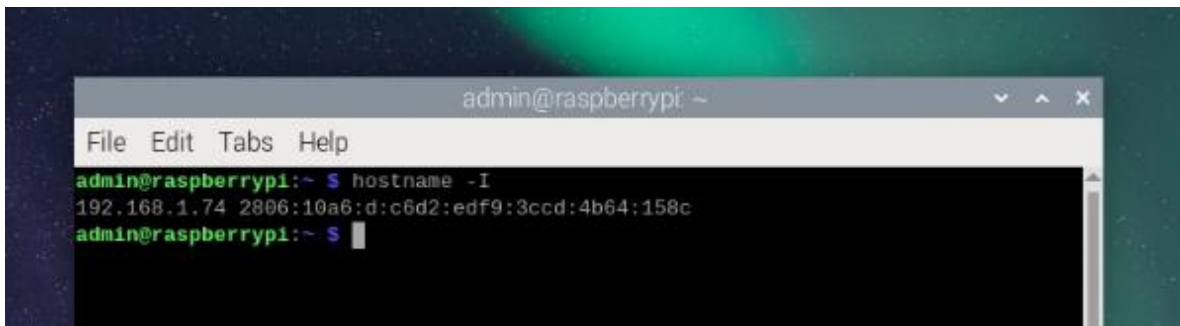
*Ilustración 21 Sitio Web WinSCP*

No se requirió la instalación de VNC, FileZilla ni navegadores adicionales, ya que el acceso completo se gestionó mediante SSH y la transferencia de archivos mediante WinSCP.

## 11. Configuración de acceso remoto

### 11.1 Dirección IP: Ejecución de hostname -I en terminal de Raspberry Pi.

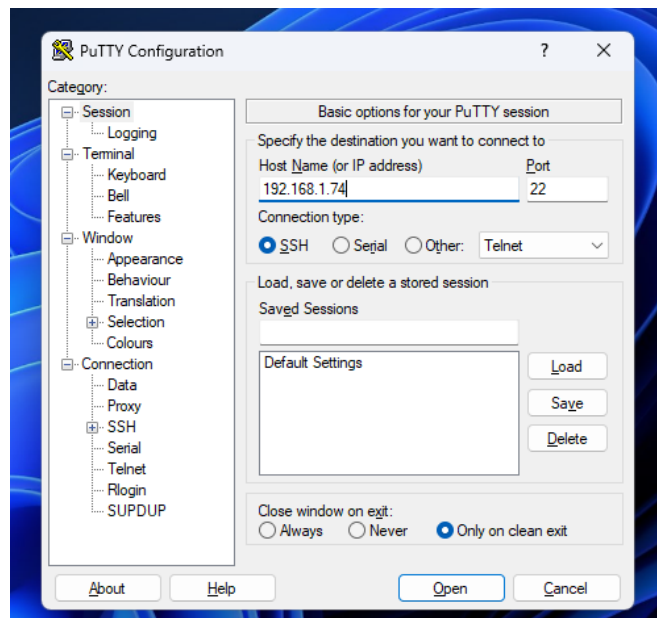
```
hostname -I
```



*Ilustración 22 Consulta de la dirección IP*

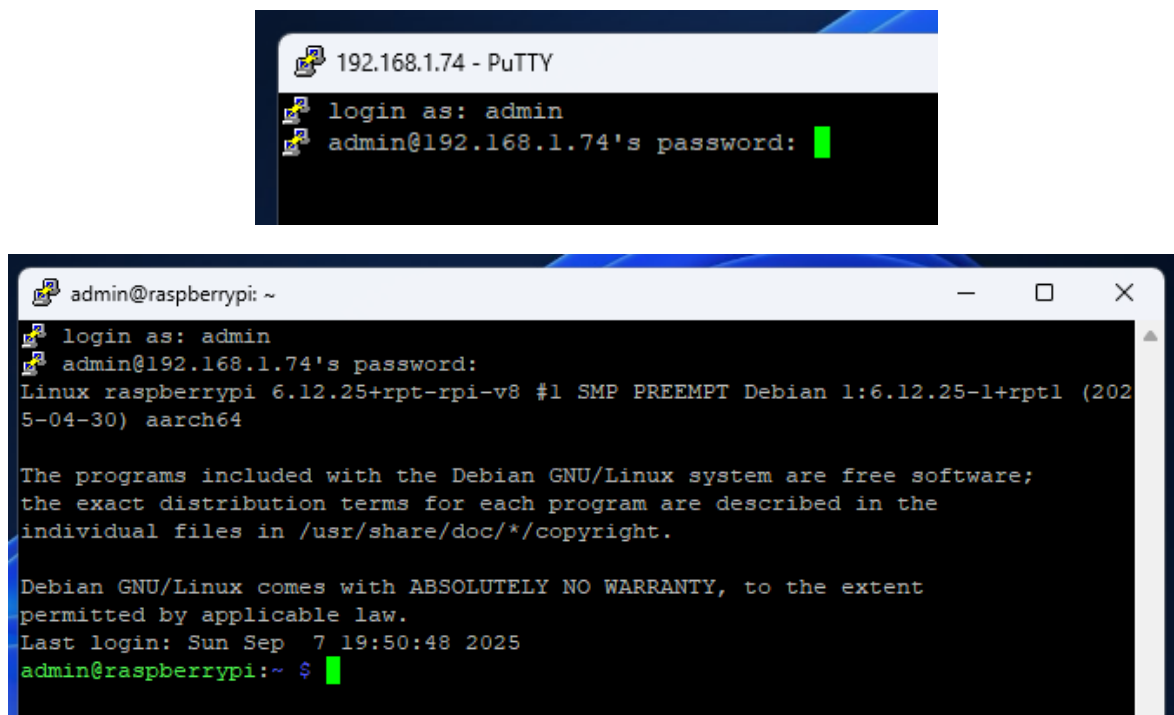
### 11.2 Conexión con PuTTY

Con PuTTY en Windows, ingresar la dirección IP de la Raspberry, seleccionar puerto 22 y conexión SSH:



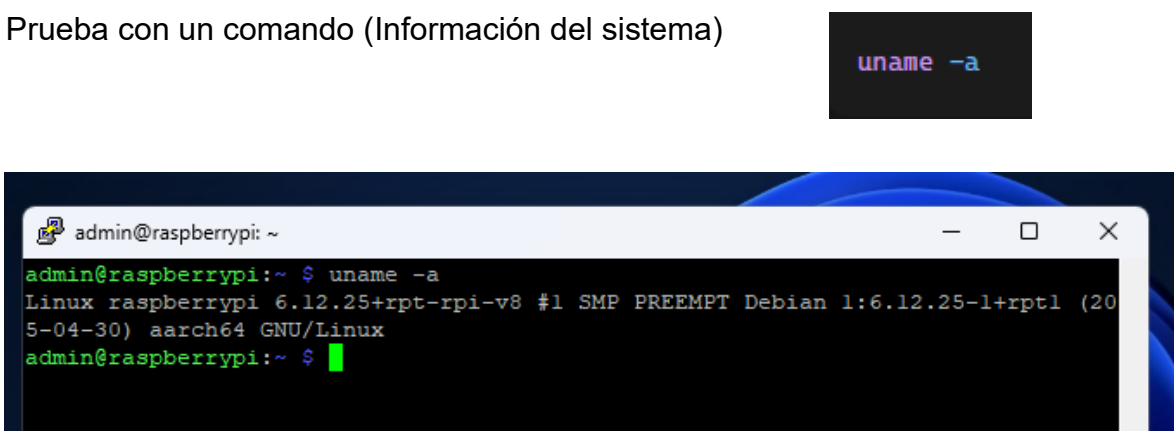
*Ilustración 23 Interfaz de PuTTY*

Ingresar credenciales previamente configuradas, posteriormente aparecerá el mensaje de conexión exitosa:



*Ilustración 24 Petición de acceso con PuTTY*

Prueba con un comando (Información del sistema)



*Ilustración 25 Consulta de Información del Sistema*

### 11.3 Conexión con WinSCP

Al tener instalado y ejecutado en Windows, WinSCP solicitará de igual manera la dirección IP de la Raspberry Pi que esté conectada a nuestra red, al igual que las credenciales que configuramos al inicio:

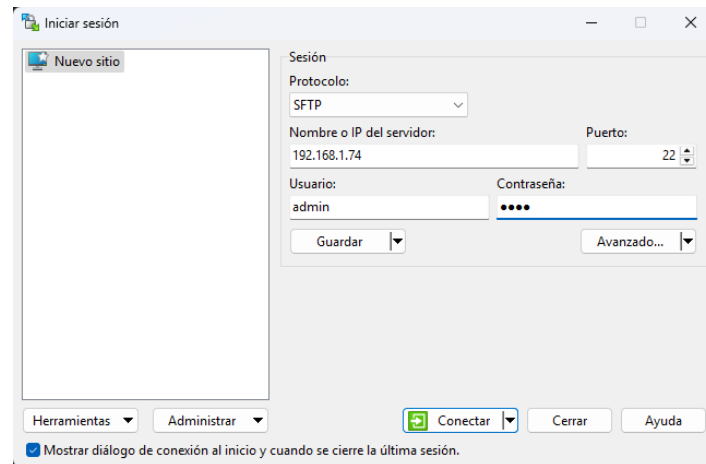


Ilustración 26 Interfaz de WinSCP

Dispondremos de la siguiente interfaz, de esta manera podremos compartir archivos entre dispositivos arrastrando y soltando dentro de la interfaz, en nuestro caso archivos .java para pruebas en Raspberry PI. (Windows lado izquierdo, Raspberry lado derecho)

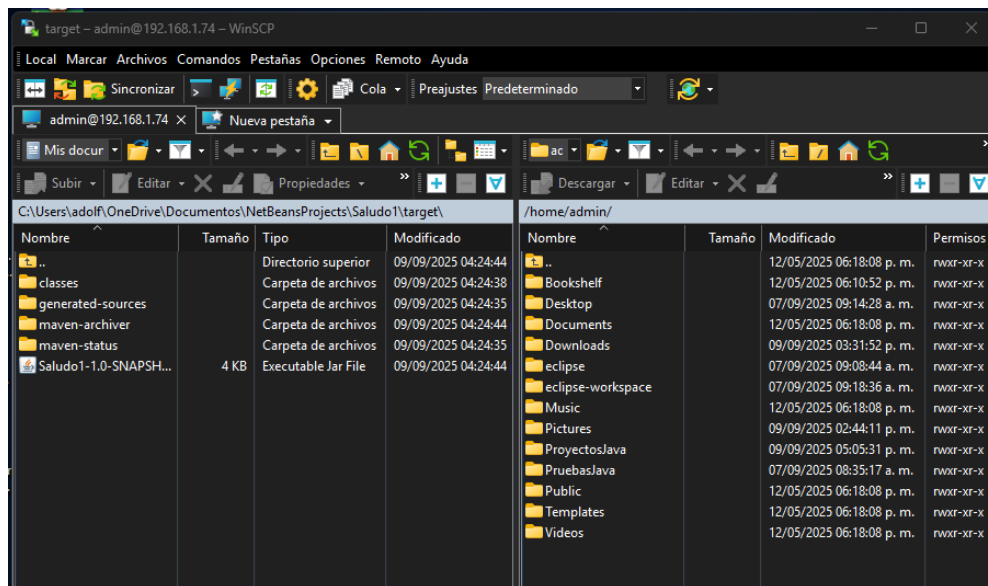


Ilustración 27 Interfaz WinSCP Entre Dispositivos

## 11.4 Integración entre entornos

Para facilitar el flujo de trabajo entre ambos entornos, se estableció el siguiente proceso:

1. **Desarrollo en Windows:** Programación, compilación y pruebas iniciales en NetBeans.
2. **Empaquetado:** Generación del archivo .JAVA y .JAR ejecutable mediante "Clean and Build" en NetBeans.
3. **Transferencia a Raspberry Pi:** Uso de WinSCP para copiar los .JAVA y .JAR desde Windows a la Raspberry Pi mediante conexión SFTP.
4. **Ejecución remota:** Acceso vía PuTTY (SSH) para ejecutar `java -jar modulo_ecuaciones.jar`.
5. **Pruebas simultáneas:** Ejecución en ambos sistemas para comparar comportamiento y visualización.

## 12. Diseño y Prototipado de interfaces

Para el diseño de las interfaces del módulo "Problemas con una variable" se utilizó Canva (versión web), una herramienta de diseño gráfico que permitió crear prototipos visuales de manera rápida e iterativa, siguiendo los principios de diseño de interfaces responsivas y usabilidad (Corcuera, 2025).

### 12.1 Proceso de diseño

El proceso siguió estas etapas:

#### 1. Análisis de requisitos visuales:

- Estudio de los módulos existentes en el sistema de álgebra lineal para mantener consistencia visual.

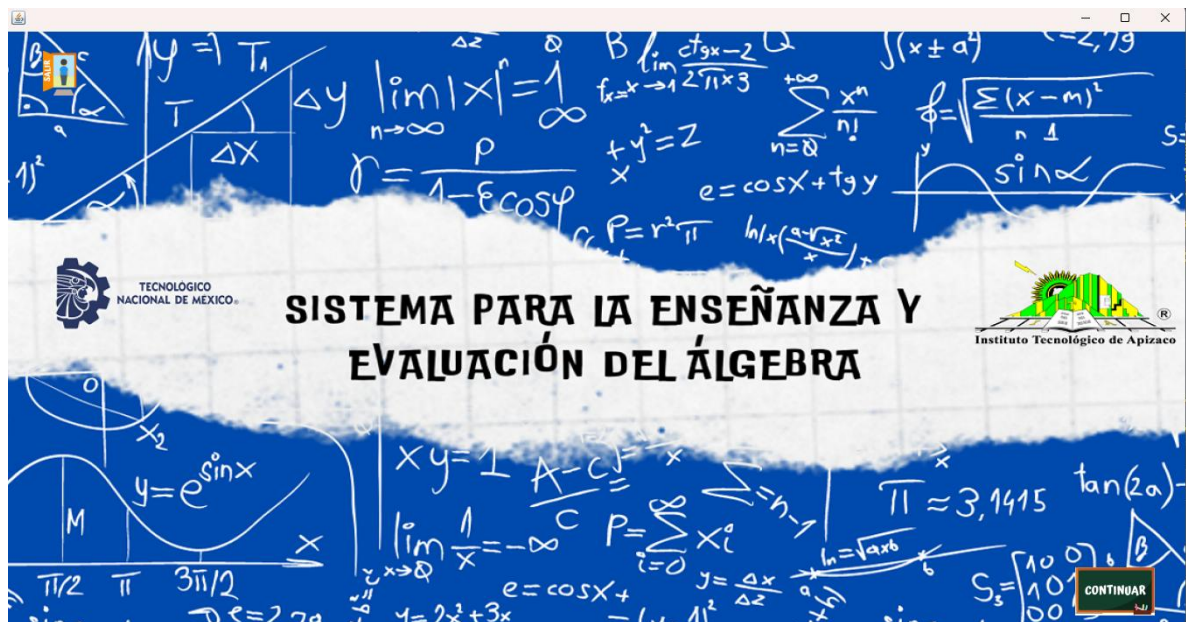


Ilustración 28 Interfaz Portada

- Definición de una paleta de colores (tonos verdes, azules y blancos).
- Establecimiento de guías de estilo: tamaños de botones, márgenes.

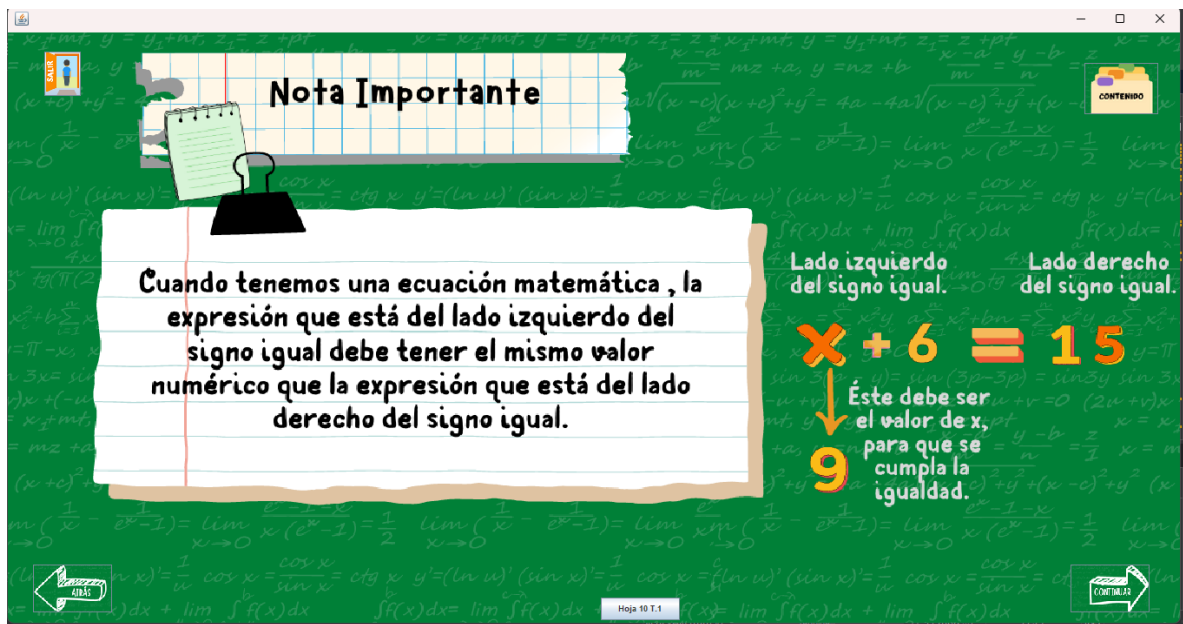


Ilustración 29 Interfaz del Tema 2

## 2. Creación de categorías de interfaces

26 interfaces se agruparon en 3 categorías principales:

| Categoría                  | Cantidad de Interfaces | Propósito                                                    |
|----------------------------|------------------------|--------------------------------------------------------------|
| Navegación principal       | 1                      | Menús, selección de temas, regreso al sistema principal      |
| Presentación de ejercicios | 24                     | Visualización de ecuaciones, área de trabajo                 |
| Interacción y evaluación   | 1                      | Ingreso de respuestas, botones de control, retroalimentación |

Ilustración 30 Categoría de Interfaces

### 3. Selección de interfaces representativas para documentación:

Para este reporte, se describen las 3 interfaces más significativas:

#### a) Interfaz principal del módulo:

- **Función:** Punto de entrada al módulo "Problemas con una variable".
- **Elementos clave:** Título del módulo, botón "Iniciar ejercicio", botón "Instrucciones", botón "Regresar al menú principal".
- **Diseño:**



Ilustración 31 Interfaz Contenido

#### b) Interfaz de ejercicio:

- **Función:** Presenta la ecuación y permite ver el proceso de resolución.
- **Elementos clave:**
  - Área de visualización de la ecuación (ej:  $2m + 24 = 106$ )
  - Botones: "Regresar", "Continuar", "Menú" y "Salir"

- **Consideraciones de diseño:** Distribución equilibrada, jerarquía visual clara.

El prototipo de interfaz de tema 4 se presenta en una barra superior azul con botones de edición, retroceso, avance, borrado y herramientas de presentación. El contenido principal está dividido en tres pantallas de una pizarra digital verde.

**Pantalla 1: Problema 3**

Un cartel naranja indica "Problema 3". A la izquierda, un cuaderno de espiral contiene el texto: "Dividir el número 106 en dos partes, tales que la mayor exceda a la menor en 24." A la derecha, un cuadro blanco con borde naranja muestra la frase: "La frase, la suma de dos números es 106, representa lo siguiente:".

| Número mayor | Suma | Número menor | Es igual | Total |
|--------------|------|--------------|----------|-------|
| $>$          | $+$  | $<$          | $=$      | 106   |
| $m$          | $+$  | $m=24$       | $=$      | 106   |

La mayor excede a la menor en 24  $\rightarrow m = 24$

**Pantalla 2: Reduciendo nos queda una ecuación con tres términos algebraicos**

Un cuaderno de espiral indica: "Para dejar solo al término de la variable, sumamos 24, en ambos lados de la ecuación". El cuadro blanco muestra la ecuación:  $2m - 24 + 24 = 106 + 24$  y  $2m = 130$ .

**Pantalla 3: Para dejar sola la variable, dividimos ambos lados de la ecuación entre 2**

El cuadro blanco muestra la ecuación:  $2m / 2 = 130 / 2$  y  $m = 65$ .

Ilustración 32 Prototipos de Interfaz del Tema 4

El diseño busca implementar pizarras blancas, ya que brindan flexibilidad en la educación y permiten al docente analizar, diseñar e implementar acciones para lograr un resultado pedagógico deseado (Specialist., 2021).

c) Interfaz de reto del Tema 4:

- **Función:** Muestra una ecuación aleatoria y si la respuesta fue correcta o incorrecta, con explicación.
- **Elementos clave:**
  - Mensaje prominente ("¡Correcto!" o "Intenta nuevamente")
  - Explicación del procedimiento



Ilustración 33 Prototipo de Interfaz Reto del Tema4

## 12.2 Principios de diseño aplicados

### 1. Consistencia visual:

- Mismos colores, y estilo de botones en todas las interfaces.
- Posicionamiento consistente de elementos de navegación.

### 2. Responsividad anticipada:

- Diseño de layouts que pudieran adaptarse a diferentes resoluciones.
- Consideración de espacios mínimos para dispositivos pequeños.

### 3. Enfoque en la experiencia educativa:

- Minimización de elementos distractores.
- Énfasis visual en la ecuación y el área de respuesta.

## 12.3 Salidas del proceso de prototipado

Como resultado de esta fase se obtuvieron:

- **25-31 archivos .png** con los diseños de cada interfaz.
- **Aprobación formal** de los diseños clave por parte de los asesores.
- **Base visual** para la implementación en Java.

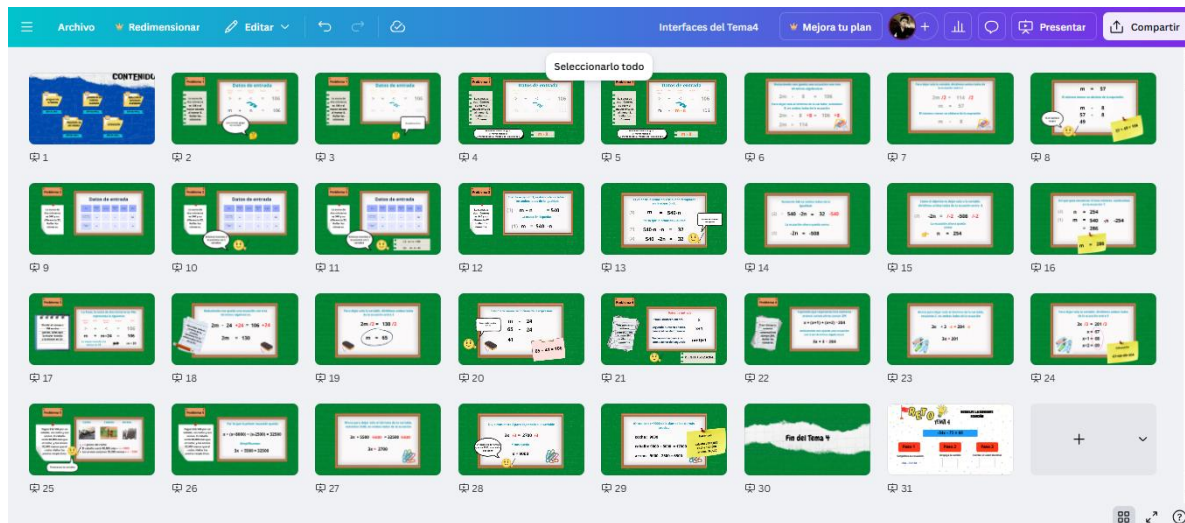


Ilustración 34 Galería de Prototipo de Interfaces

## 13. Programación de las interfaces gráficas

### 13.1 Preparación y procesamiento de recursos gráficos

Antes de cualquier implementación en código, fue necesario realizar un proceso exhaustivo de preparación visual de todos los elementos de interfaz. Este proceso garantizó que las imágenes se integraran perfectamente en la aplicación, sin artefactos visuales ni problemas de superposición.

#### Herramientas y técnicas utilizadas:

1. **Remove.bg** (Sitio web <https://www.remove.bg/es>):
  - Remoción de fondos blancos mediante varita mágica y tijeras inteligentes
  - Exportación en formato **PNG** para preservar transparencia
2. **Proceso manual de limpieza:**
  - Pixelado de bordes para eliminar halos blancos residuales
3. **Optimización para Java:**
  - Reducción de tamaño sin pérdida de calidad visual
  - Organización en estructura de directorios lógica
  - Nomenclatura camelCase: tipoElementoNumero.png

```
// LOGO PROBLEMA - FIJO (sin animación)
int logoAncho = (int) (a11 * PROPORCION_LOGO_ANCHO);
int logoAlto = (int) (a22 * PROPORCION_LOGO_ALTO);
logoProblema = crearImagen("/imagenes4/logoProblema3.png", logoAncho, logoAlto);

// Posición logo (centrado en la parte superior)
int logoX = (int) (a11 * 0.10); // PORCENTAJE HORIZONTAL (0.10 = 10% desde izquierda)
int logoY = (int) (a22 * 0.01); // PORCENTAJE VERTICAL DESDE ARRIBA
logoProblema.setLocation(logoX, logoY);
logoProblema.setVisible(true); // Visible desde el inicio

// NOTA PROBLEMA
int notaAncho = (int) (a11 * PROPORCION_NOTA_ANCHO);
int notaAlto = (int) (a22 * PROPORCION_NOTA_ALTO);
notaProblema = crearImagen("/imagenes4/notaProblema3.png", notaAncho, notaAlto);
```

Ilustración 35 Llamado de imágenes

## Impacto en la implementación:

1. **Superposiciones limpias:** Los elementos se apilan sin bordes blancos
2. **Animaciones fluidas:** Las transiciones no muestran artefactos
3. **Responsividad efectiva:** El escalado no revela fondos no deseados
4. **Experiencia inmersiva:** Interfaz visualmente coherente

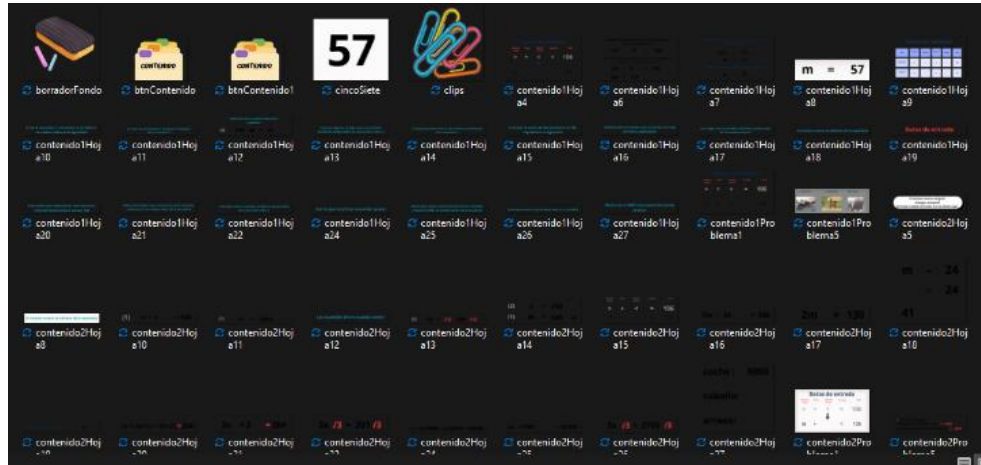


Ilustración 36 Imágenes recortadas

## 13.2 Patrón de desarrollo

Mientras que el sistema base utilizaba un enfoque de JButtons con imágenes estáticas y responsividad mediante divisiones enteras, para el módulo "Problemas con una variable" se implementó una evolución significativa que incluyó:

1. **Animaciones de componentes** (fade in, deslizamientos, efectos intermitentes)
2. **Responsividad basada en porcentajes** (más preciso y mantenible)
3. **Gestión avanzada de capas visuales** (orden Z explícito)
4. **Coordinación temporal de elementos** mediante múltiples timers

Se conservo el fondo que incluyen el resto de las interfaces que contienen los otros temas (color verde que muestra un pizarrón con símbolos matemáticos)

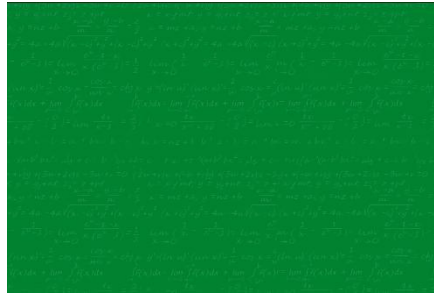


Ilustración 37 Fondo Pizarra Verde

### 13.3 Sistema de animaciones

Todas interfaces que forman parte del Contenido 4 del Sistema cuentan en su mayoría con al menos dos de tres tipos de animaciones aplicadas a los elementos gráficos siendo las siguientes:

#### a) Animaciones de entrada (slide-in):

Cómo se menciona anteriormente todas las interfaces siguen un diseño específico el cual es que, al inicio de cada problema, este se describe en una nota, después aparezca una pizarra y dentro de ella se desarrolle el procedimiento de solución, por lo que en este caso, la nota del problema es a la que se le aplica este método de animación `iniciarAnimacionNotaProblema`, la cual aparece desde el lado izquierdo de la pantalla (Design, 2023):

```
private void iniciarAnimacionNotaProblema() {  
    // Posición final (ya calculada)  
    final int posFinalX = notaProblema.getX();  
    final int posFinalY = notaProblema.getY();  
  
    // Posición inicial (fuera de pantalla a la izquierda)  
    notaProblema.setLocation(-notaProblema.getWidth(), posFinalY);  
    notaProblema.setVisible(true);  
  
    Timer timerNota = new Timer(20, new ActionListener() {  
        int posXActual = -notaProblema.getWidth();  
        private final int velocidad = (int) (all * 0.05);  
  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            posXActual += velocidad; //CAMBIO: velocidad responsiva
```

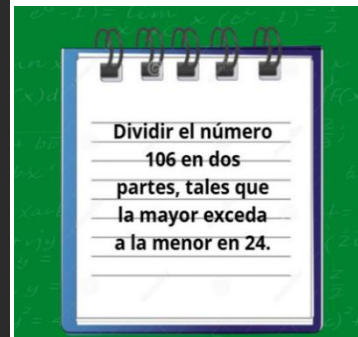


Ilustración 38 Ejemplo Método `iniciarAnimacionNotaProblema`

## b) Efectos de transparencia (fade in):

igual que al caso anterior se buscó utilizar una animación llamativa pero diferente a la anterior por lo que se optó por aplicar la animación fade-in, la cual consiste en mostrar un elemento el cual va apareciendo de forma transparente hasta volverse completamente visible, como lo es en el caso del contenido que se muestra en la pizarra que inicialmente se mostrará vacía (Design, 2023):

```
private void iniciarFadeIn(JLabel contenido) {
    contenido.setVisible(true);
    ImageIcon imagenOriginal = (ImageIcon) contenido.getIcon();
    Image imagenCompleta = imagenOriginal.getImage();

    javax.swing.JPanel panelFade = new javax.swing.JPanel() {
        float alpha = 0f;
        @Override
        protected void paintComponent(java.awt.Graphics g) {
            super.paintComponent(g);
            java.awt.Graphics2D g2d = (java.awt.Graphics2D) g.create();
            g2d.setComposite(java.awt.AlphaComposite.getInstance(java.awt.
            g2d.drawImage(imagenCompleta, 0, 0, getWidth(), getHeight(),
            g2d.dispose();
        }
        void setAlpha(float value) {
            alpha = Math.min(Math.max(0f, value));
            repaint();
        }
    };
};
```

```
@Override
public void actionPerformed(ActionEvent e) {
    switch (etapa) {
        case 0:
            // CONTENIDO 1 - FADE IN
            iniciarFadeIn(contenido1);
            break;
        case 1:
            // CONTENIDO 2 - FADE IN
            iniciarFadeIn(contenido2);
            break;
        case 2:
            // CONTENIDO 3 - FADE IN
            iniciarFadeIn(contenido3);
            break;
        case 3:
            // CONTENIDO 4 - NORMAL (sin animación)
            contenido4.setVisible(true);
            break;
        case 4:
            // CONTENIDO 5 - NORMAL (sin animación)
            contenido5.setVisible(true);
            break;
    }
}
```

**La frase, la suma de dos números es 106, representa lo siguiente:**

| Número mayor | Suma | Número menor | Es igual | Total |
|--------------|------|--------------|----------|-------|
| >            | +    | <            | =        | 106   |
| m            | +    | m=24         | =        | 106   |

**La mayor excede a la menor en 24** → **m = 24**

Ilustración 40Ejemplo de Método iniciarFadeIn

### c) Efectos intermitentes:

Otra animación que se mantiene en la mayoría de las interfaces es la intermitencia de elementos como números que indican las operaciones que se deben aplicar al simplificar una ecuación o despejar las variables (Mayer, 2002).

```
private void iniciarIntermitenteMas24() {  
    if (timerIntermitente != null && timerIntermitente.isRunning()) {  
        timerIntermitente.stop();  
    }  
  
    timerIntermitente = new Timer(500, new ActionListener() {  
        boolean visible = true;  
        int contador = 0;  
        final int DURACION_SEGUNDOS = 8;  
        final int CICLOS_TOTALES = (DURACION_SEGUNDOS * 1000) / 500;  
    });  
}
```

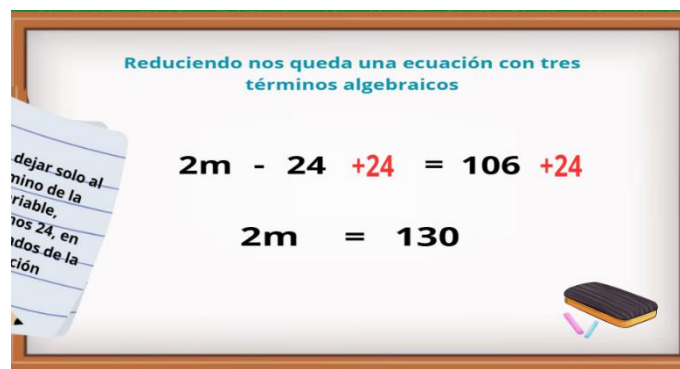


Ilustración 41 Ejemplo Método iniciarIntermitentes

## 13.4 Responsividad avanzada basada en porcentajes

En base a las divisiones enteras ( $a1/32$ ,  $a2/14$ ), se implementó un sistema más flexible basado en porcentajes y en algunos casos en referencia a algunos elementos para mantener las posiciones totalmente relativas (M, 2020).

```
// CONSTANTES DE PROPORCIÓN RESPONSIVAS  
final double PROPORCION_LOGO_ANCHO = 0.20;  
final double PROPORCION_LOGO_ALTO = 0.15;  
final double PROPORCION_NOTA_ANCHO = 0.25;  
final double PROPORCION_NOTA_ALTO = 0.60;  
final double PROPORCION_PIZARRA_ANCHO = 0.55;  
final double PROPORCION_PIZARRA_ALTO = 0.60;
```

```
// NOTA PROBLEMA
int notaAncho = (int)(all * PROPORCION_NOTA_ANCHO);
int notaAlto = (int)(a22 * PROPORCION_NOTA_ALTO);
notaProblema = crearImagen("/imagenes4/notaProblema3.png", notaAncho, notaAlto);

// Posición nota (lado izquierdo)
int notaX = (int)(all * 0.05);
int notaY = (int)(a22 * 0.20);
notaProblema.setLocation(notaX, notaY);
notaProblema.setVisible(false); // Se mostrará con animación

// PIZARRA VACIA - AL LADO DE NOTA PROBLEMA
int pizarraAncho = (int)(all * PROPORCION_PIZARRA_ANCHO);
int pizarraAlto = (int)(a22 * PROPORCION_PIZARRA_ALTO);
pizarraVacía = crearImagen("/imagenes4/pizarraVacíaBlanca.png", pizarraAncho, pizarraAlto);

// Posición pizarra (lado derecho, al lado de la nota)
int pizarraX = notaX + notaAncho + (int)(all * 0.06); // Separación de 6%
int pizarraY = (int)(a22 * 0.20);
pizarraVacía.setLocation(pizarraX, pizarraY);
pizarraVacía.setVisible(false); // Se mostrará con animación
```

*Ilustración 42 Ejemplo de cálculos responsivos*

### 13.5 Gestión de capas visuales (orden Z)

Para evitar superposiciones incorrectas, se implementó un sistema explícito de orden Z a todas las interfaces ya que algunas veces el no aplicarlo generaba inconsistencias demasiado visibles:

```
// ORDEN Z EXPLÍCITO
etqPortada.setComponentZOrder(contenido6, 0);
etqPortada.setComponentZOrder(contenido5, 1);
etqPortada.setComponentZOrder(contenido4, 2);
etqPortada.setComponentZOrder(contenido3, 3);
etqPortada.setComponentZOrder(contenido2, 4);
etqPortada.setComponentZOrder(contenido1, 5);
etqPortada.setComponentZOrder(pizarraVacía, 6);
etqPortada.setComponentZOrder(notaProblema, 7);
etqPortada.setComponentZOrder(logoProblema, 8);
```

*Ilustración 43 Ejemplo de Orden Z Explícito*

### 13.6 Estructura de recursos específica para el tema 4

SiEnEvAI0 (Proyecto) que hasta el momento contiene 4 carpetas, imagenes / imagenes2 contienen las imágenes de los contenidos de los Temas 1 y 2 del sistema, así que nuestras imágenes que utilizamos para el desarrollo del Tema 4 se encuentran en la carpeta imagenes4, y en sieneval0 contiene las clases de código que componen el sistema.

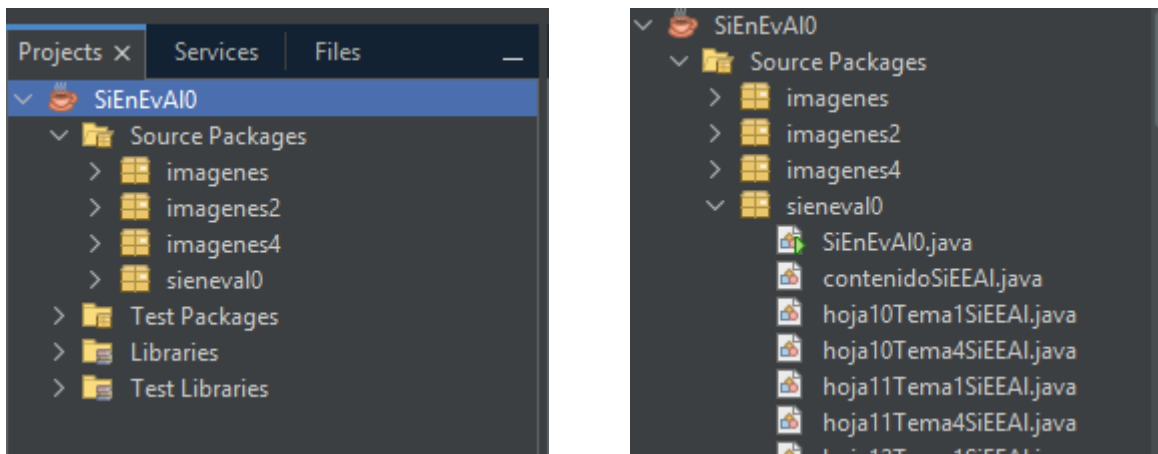


Ilustración 44 Estructura del Proyecto

Todos los elementos gráficos (imágenes), de todas las clases se inician por medio del método inicializarElementosGraficos:

```
private void inicializarElementosGraficos() {
    // CONSTANTES DE PROPORCIÓN RESPONSIVAS
    final double PROPORCION_LOGO_ANCHO = 0.20;
    final double PROPORCION_LOGO_ALTO = 0.15;
    final double PROPORCION_NOTA_ANCHO = 0.25;
    final double PROPORCION_NOTA_ALTO = 0.60;
    final double PROPORCION_PIZARRA_ANCHO = 0.55;
    final double PROPORCION_PIZARRA_ALTO = 0.60;

    // LOGO PROBLEMA - FIJO (sin animación)
    int logoAncho = (int) (a11 * PROPORCION_LOGO_ANCHO);
    int logoAlto = (int) (a22 * PROPORCION_LOGO_ALTO);
    logoProblema = crearImagen("/imagenes4/logoProblema3.png", logoAncho, logoAlto);

    // Posición logo (centrado en la parte superior)
    int logoX = (int) (a11 * 0.10); // PORCENTAJE HORIZONTAL (0.10 = 10% desde izquierda)
    int logoY = (int) (a22 * 0.01); // PORCENTAJE VERTICAL DESDE ARRIBA
    logoProblema.setLocation(logoX, logoY);
    logoProblema.setVisible(true); // Visible desde el inicio
}
```

Ilustración 45 Ejemplo Método inicializarElementos de una clase

## 14. Programación del algoritmo para la generación de ecuaciones y su solución

### 14.1 Análisis del algoritmo base proporcionado

Al inicio del proyecto, se recibió un algoritmo base escrito en Java que generaba ecuaciones algebraicas aleatorias para el Reto del Tema 1. Sin embargo, este algoritmo presentaba varias limitaciones para el propósito específico del módulo "Problemas con una variable":

#### Problemas identificados en el algoritmo original:

1. **Generación con exponentes:** Producía ecuaciones del tipo  $AX^2 + BX + C = D$
2. **Visualización incorrecta:** Los exponentes no se mostraban correctamente en la interfaz.
3. **Campos vacíos:** Algunas cajas se mostraban vacías sin ecuación

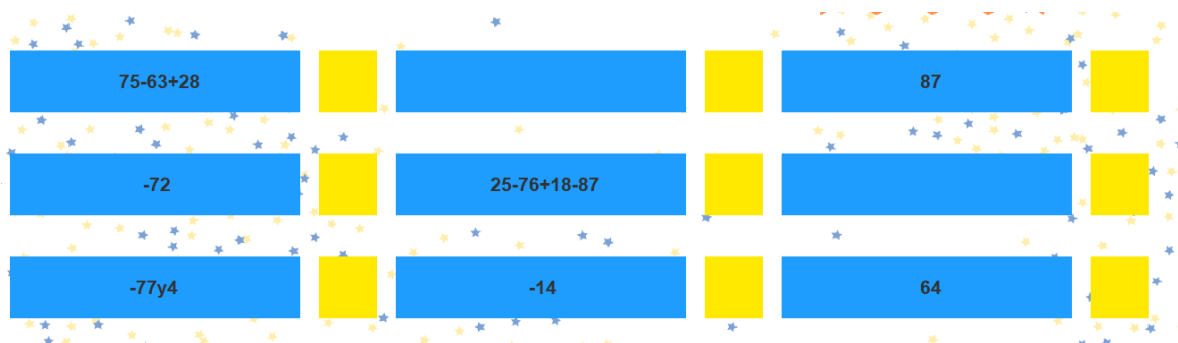


Ilustración 46 Reto Tema 1 inconsistente

### 14.2 Proceso de solución al algoritmo

- El primer paso fue aplicar estilo CSS que mejora el centrado y vista de expresiones en el método *convierteExpon* (Lenguajecss, 2025).

```
public static String convierteExpon(String cad1)

StringBuilder sb = new StringBuilder("<html><body style='font-family:Arial; font-size:18px;'>");
sb.append(cad1);
```

Ilustración 47 Método *convierteExpon* ANTES

```

public static String convierteExpon(String cad1)
{
    // CSSS QUE MEJORA EL CENTRADO COMPLETO (horizontal y vertical)
    StringBuilder sb = new StringBuilder(
        "<html><body style='" +
        "font-family: Arial; " +
        "font-size: 20px; " +           //ORIGINALMENTE ESTABA EN 18PX
        "font-weight: bold; " +
        "margin: 0; " +
        "padding: 0; " +
        "display: flex; " +             // ACTIVA FLEXBOX
        "align-items: center; " +       // ENTRADO VERTICAL
        "justify-content: center; " +   // CENTRADO HORIZONTAL
        "height: 100%; " +             // OCUPA TODO EL ALTO
        "width: 100%; " +              // OCUPA TODO EL ANCHO
        "text-align: center; " +       // CENTRADO EXTRA
        ">"
    );
}

```

*Ilustración 49 Método convierteExpon DESPUES*

- El segundo cambio fue dejar solo un componente de texto ya que existían que se superponían, además de que solo se buscaba trabajar solo con JTextPane de preferencia, por lo que se eliminó el JTextField:

```

void colocaCjsTxt(int x, int y, int an, int al, int color) {
    JTextField txt=new JTextField();
    JTextPane txt2= new JTextPane();
    int c1=0, c2=0, c3=0; //Son los códigos de los colores
    if (color ==1) //color= 1 es rosa

```

*Ilustración 50 Método colocaCjsTxt ANTES*

```

void colocaCjsTxt(int x, int y, int an, int al, int color) {

    JTextPane txt = new JTextPane(); //DEJO UN SOLO COMPONE

```

*Ilustración 51 Método colocaCjsTxt DESPUES*

- Lo siguiente fue agregar una validación para las cajas que se detecten vacías, y de esa manera se le asigne una ecuación sin que quede alguna vacía

```

//EVITAR CAJAS VACÍAS - VERIFICACIÓN EXTRA
if (s2 == null || s2.trim().isEmpty() || s2.equals(" ") || s2.length() < 2) {
    System.out.println("Expresión vacía detectada, generando una por defecto");
    s2 = "3x^2 + 2x + 1"; // Expresión de respaldo
}

```

*Ilustración 52 Condición SIN CAJAS VACIAS*

De esta manera queda resulta la parte que se mantenía inconsistente en cuanto a la visualización de las expresiones aleatorias del Reto del Tema 1:

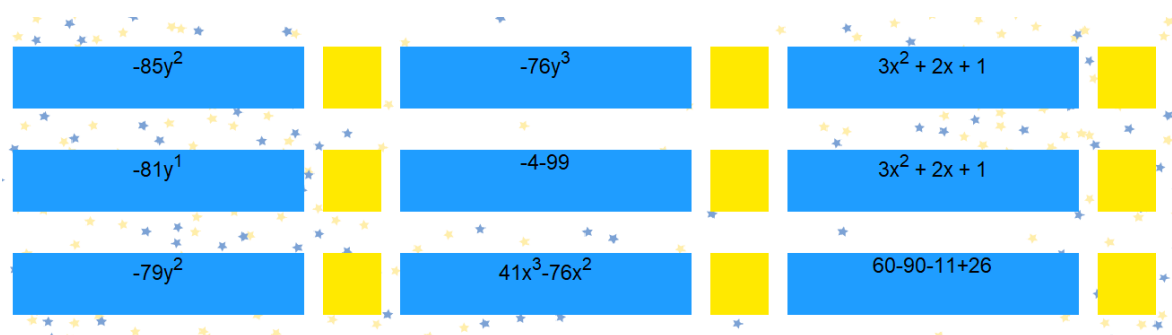


Ilustración 53 Reto Tema 1 Mejorado

### 14.3 Proceso de adaptación y simplificación

Convertir un generador de expresiones algebraicas complejas en un generador de ecuaciones lineales didácticas para el módulo "Problemas con una variable".

- Lo primero es extraer la base del algoritmo original, pero lo simplificamos:

```
// ORIGINAL:
int vEnt1 = numAleatorio(100); // Coeficiente 0-99
int vEnt2 = numAleatorio(5); // Exponente 0-5
String s3 = (varXY==0) ? "x" : "y"; // Variable aleatoria
s2 = s1 + cadEnt + s3 + "^" + s4; // Construcción compleja
```

Ilustración 54 Base de algoritmo original

```
// ADAPTADO:
int coefX = numAleat.nextInt(99) + 1; // 1-99 (sin cero)
// Exponente FIJO = 1 (implícito, no se muestra)
String variable = (numAleat.nextBoolean()) ? "x" : "y"; // Variable consistente
```

Ilustración 55 Adaptación nueva del algoritmo

## Eliminación del método de conversión

Eliminamos el método de conversión por el tema de ecuaciones de primer grado (sin exponentes)

```
public static String convierteExpon(String cad1)

StringBuilder sb = new StringBuilder("<html><body style='font-family:Arial; font-size:18px;'>");
sb.append(cad1);
```

*Ilustración 56 Método convierteExpon ELIMINADO*

```
private String generarEcuacionPrimerGrado() {
    int coefX = numAleat.nextInt(99) + 1;
    int terminoIndep = numAleat.nextInt(99) + 1;
    int constante = numAleat.nextInt(99) + 1;

    String signol = (numAleat.nextBoolean()) ? "+" : "-";
    String signo2 = (numAleat.nextBoolean()) ? "+" : "-";

    String variable = (numAleat.nextBoolean()) ? "x" : "y";
```

*Ilustración 57 NUEVO Método generarEcuaciónPrimerGrado*

Se genera el método *extraerCoeficientes* que tomará los valores a los cuales se les tendrán que aplicar los cálculos:

```
// EXTRAER COEFICIENTES a, b, c
private void extraerCoeficientes(String ecuacion) {
    ecuacion = ecuacion.replace(" ", "");
    String[] partes = ecuacion.split("=");
    String ladoIzq = partes[0];
    constanteC = Double.parseDouble(partes[1]);

    variable = ladoIzq.contains("x") ? "x" : "y";

    String ladoIzqTemp = ladoIzq;
    String strA = "";
    String strB = "";

    int posVariable = ladoIzqTemp.indexOf(variable);

    if (posVariable > 0) {
        strA = ladoIzqTemp.substring(0, posVariable);
    } else if (posVariable == 0) {
        strA = "1";
    }
}
```

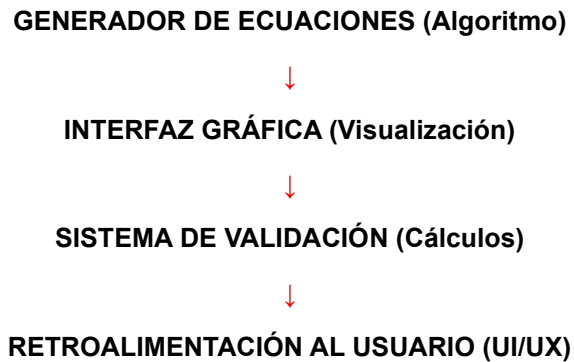
*Ilustración 58 Método extraerCoeficientes*

### **Ventajas de la adaptación**

1. **Enfoque pedagógico:** Ecuaciones adecuadas para **principiantes** en álgebra
2. **Robustez matemática:** Coeficientes nunca cero  $\rightarrow$  siempre divisible
3. **Integración perfecta:** Generación + extracción + indicaciones en el flujo
4. **Mantenibilidad:** Código más simple, menos propenso a errores

## 15. Integración del algoritmo en las interfaces

La integración sigue un flujo secuencial donde cada componente alimenta al siguiente:



### 15.1 Punto de integración principal: int1Ret1T4SiEEAI

La clase int1Ret1T4SiEEAI (Reto del Tema 4), es el corazón de la integración, donde convergen todos los componentes:

```
public class int1Ret1T4SiEEAI extends javax.swing.JFrame {  
    JLabel etqPortada = new JLabel();  
    private JButton btnSalir = new JButton();  
    private JButton btnContenido = new JButton();  
  
    // Variables para tamaño de pantalla  
    int anchoPantalla, altoPantalla;
```

*Ilustración 58 Clase int1Ret1T4SiEEAI*

## Secuencia de integración en el constructor

De igual manera se agregan los elementos que conformaran la interfaz, así como en las interfaces anteriores, agregando los elementos gráficos, integración del algoritmo y la implementación de una nueva clase llamada confeti, la cual mostrara una animación de festejo si se resuelve la ecuación completa

```
// Colocar imagen de fondo
String ligaImg = "/imagenes4/imgEv4Blanco.png";
colocaImgEnFrame(ligaImg, al, a2);

// INICIALIZAR SISTEMA DE CONFETI
confeti = new SistemaConfetiSimple();
confeti.setBounds(0, 0, al, a2);
etqPortada.add(confeti);

// Agregar componentes
this.add(etqPortada);

// INICIAR SISTEMA PASO A PASO
iniciarSistemaPasos(al, a2);

this.add(etqPortada);
}
```

*Ilustración 59 int1Ret1T4SiEEA1*

## Método iniciarSistemaPasos() - Núcleo de la integración

En este método integra todo el proceso de evaluación, llamando al algoritmo, lo muestra, aplica los cálculos, y valida mostrando mensajes:

```
// MÉTODO PRINCIPAL MODIFICADO
private void iniciarSistemaPasos(int ancho, int alto) {
    // 1. Generar ecuación aleatoria
    ecuacionGenerada = generarEcuacionPrimerGrado();

    // 2. Mostrar ecuación inicial
    mostrarEcuacionInicial(ancho, alto);

    // 3. Extraer coeficientes a, b, c
    extraerCoeficientes(ecuacionGenerada);

    // 4. Calcular resultados anticipados
    resultadoPasol = constanteC - terminoB;
    solucionFinal = resultadoPasol / coeficienteA;

    // 5. Iniciar el paso 1
    iniciarPasol(ancho, alto);
}
```

*Ilustración 60 Método iniciarSistemaPasos*

## Integración del algoritmo en la visualización

Se crea el método *mostrarEcuacionInicial* el cual se encarga de generar y mostrar en la interfaz la ecuación aleatoria dentro de un JTextPane con color y tamaño específicos:

```
private void mostrarEcuacionInicial(int ancho, int alto) {  
    txtEcuacionDisplay = new JTextPane();  
    txtEcuacionDisplay.setContentType("text/html");  
    txtEcuacionDisplay.setEditable(false);  
  
    int txtAncho = ancho * 3 / 10;  
    int txtAlto = alto / 15;  
    int txtX = (ancho - txtAncho) / 2;  
    int txtY = alto / 3;  
  
    txtEcuacionDisplay.setBounds(txtX, txtY, txtAncho, txtAlto);  
    //txtEcuacionDisplay.setBackground(new Color(31, 157, 255));  
    txtEcuacionDisplay.setBackground(new Color(102, 255, 102));  
    txtEcuacionDisplay.setBorder(null);  
}
```

Ilustración 61 Método mostrarEcuacionInicial

## Integración en el sistema indicaciones paso a paso

Para validar los resultados de los pasos por medio de los cálculos se crean 3 métodos para los 3 pasos que se piden “Simplificar la ecuación” “Despejar la variable” y “Escribir el resultado decimal”

```
private void verificarPasos() {  
    try {  
        double inputIzq = Double.parseDouble(txtInputIzqPl.getText());  
        double inputDer = Double.parseDouble(txtInputDerPl.getText());  
  
        double valorEsperado = -terminoB;  
        boolean correcto = (Math.abs(inputIzq - valorEsperado) < 0.001 &&  
            Math.abs(inputDer - valorEsperado) < 0.001);  
  
        if (correcto) {  
            lblResultadoPl.setText(";Correcto!");  
            lblResultadoPl.setForeground(Color.GREEN);  
  
            txtInputIzqPl.setEnabled(false);  
            txtInputDerPl.setEnabled(false);  
        }  
    }  
}
```

```
private void verificarPaso2() {
    try {
        String inputIzq = txtInputIzqP2.getText().trim();
        String inputDer = txtInputDerP2.getText().trim();

        String inputIzqLimpio = inputIzq.replace("/", "").replace("÷", "").trim();
        String inputDerLimpio = inputDer.replace("/", "").replace("÷", "").trim();

        double valorIzq = Double.parseDouble(inputIzqLimpio);
        double valorDer = Double.parseDouble(inputDerLimpio);

        boolean correcto = (Math.abs(valorIzq - coeficienteA) < 0.001
            && Math.abs(valorDer - coeficienteA) < 0.001);

        if (correcto) {
            lblResultadoP2.setText("¡Correcto!");
            lblResultadoP2.setForeground(Color.GREEN);

            txtInputIzqP2.setEnabled(false);
            txtInputDerP2.setEnabled(false);
        }
    }
}
```

```
// Mensaje de éxito
lblResultadoP3.setText("<html><center><b>¡CORRECTO!</b><br>"
    + variable + " = " + String.format("%.2f", solucionFinal) + "</center></html>");
lblResultadoP3.setFont(new java.awt.Font("Comic Sans MS", java.awt.Font.BOLD,
    Math.max(16, altoPantalla/50)));
lblResultadoP3.setForeground(new Color(0, 150, 0));

txtInputP3.setEnabled(false);

// ¡DISPARAR CONFETI!
confeti.dispararConfeti(300); // 150 partículas de confeti

// Mensaje adicional de celebración
JLabel lblCelebracion = new JLabel("¡Ecuación resuelta correctamente!");
```

*Ilustración 62 Métodos de Verificación*

## Integración de animación de confeti

Una vez que se verifique el paso 3 como CORRECTO iniciara la animación confeti basada en la gamificación que aporta un valor extra positivo al sistema según (Panella, 2023).

```
private void verificarPaso3() {
    try {
        double respuesta = Double.parseDouble(txtInputP3.getText());
        double solucionRedondeada = Math.round(solucionFinal * 100.0) / 100.0;
        double respuestaRedondeada = Math.round(respuesta * 100.0) / 100.0;

        boolean correcto = (Math.abs(respuestaRedondeada - solucionRedondeada) < 0.001);

        if (correcto) {
            int anchoMensaje = anchoPantalla * 8 / 30;

            // Mensaje de éxito
            lblResultadoP3.setText("<html><center><b>¡CORRECTO!</b><br>"
                + variable + " = " + String.format("%.2f", solucionFinal) + "</center></html>");
            lblResultadoP3.setFont(new java.awt.Font("Comic Sans MS", java.awt.Font.BOLD,
                Math.max(16, altoPantalla/50)));
        }
    }
}
```

## Manejo de errores y casos límite

Se integran validaciones en los métodos de verificarPaso1, verificarPaso2 y verificarPaso3 dentro de los condicionales para una mayor robustez:

```
    } else {  
        lblResultadoP3.setText("Incorrecto");  
        lblResultadoP3.setForeground(Color.RED);  
    }  
} catch (NumberFormatException e) {  
    lblResultadoP3.setText("Ingresa un número<br>decimal válido");  
    lblResultadoP3.setForeground(Color.RED);  
}
```

```
    } else {  
        lblResultadoP2.setText("Incorrecto. Vuelve a intentarlo");  
        lblResultadoP2.setForeground(Color.RED);  
    }  
} catch (NumberFormatException e) {  
    lblResultadoP2.setText("Llena ambos campos");  
    lblResultadoP2.setForeground(Color.RED);  
}
```

Ilustración 63 Validaciones en verificarPasos



Ilustración 64 Primera Vista de la Intrfaz Reto T4

### **Ventajas de esta arquitectura integrada**

1. **Desacoplamiento moderado:** Cada componente es independiente, pero se integra fácilmente
2. **Flujo de datos unidireccional:** Algoritmo > Interfaz > Validación > Retroalimentación
3. **Reusabilidad:** El algoritmo generador podría usarse en otras interfaces
4. **Mantenibilidad:** Cambios en un componente no rompen los otros
5. **Testabilidad:** Cada componente puede probarse por separado

## 16. Pruebas de funcionamiento de las interfaces

### 16.1 Estrategia de pruebas implementada

Se realizaron pruebas en tres niveles para garantizar el correcto funcionamiento:

1. **Pruebas unitarias:** Componentes individuales
2. **Pruebas de integración:** Flujos completos
3. **Pruebas de sistema:** Comportamiento general

### 16.2 Pruebas del algoritmo generador de ecuaciones

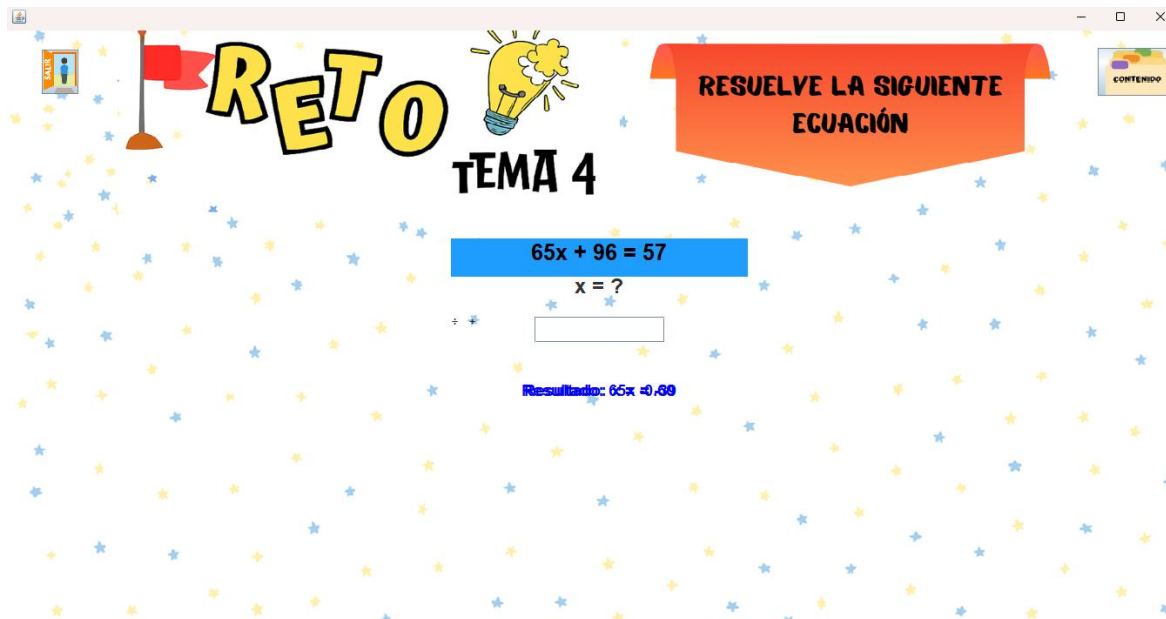


Ilustración 65 Prueba de algoritmo Reto Tema 4

Generación de ecuaciones para verificar el formato  $AX + B$ , extracción y cálculo para las ecuaciones aleatorias funcionando correctamente.

## 16.3 Pruebas de la interfaz gráfica

### a) Pruebas de renderizado en diferentes resoluciones:

#### RESOLUCIONES PROBADAS:

1. 1366×768 (Laptops estándar)
2. 1280×720 (HD básico)
3. 1920×1080 (Full HD)
4. 800×600 (Modo de compatibilidad)

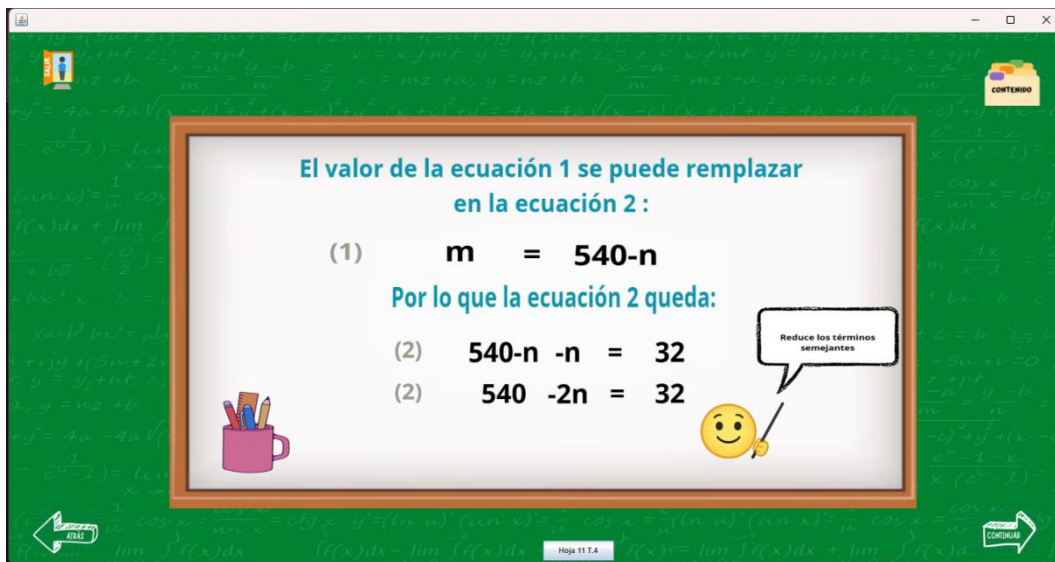


Ilustración 66 Prueba Hoja 11 Tema 4

#### VERIFICACIONES POR RESOLUCIÓN VALIDADAS:

- Todos los componentes visibles
- Textos legibles (sin superposición)
- Botones accesibles
- Campos de texto correctamente dimensionados

## b) Pruebas de las animaciones:

### ANIMACIONES VALIDADAS:

1. Slide-in de notas de problemas
2. Slide-in de pizarras
3. Fade-in en contenidos
4. Intermitencia en contenidos
5. Sistema de confeti

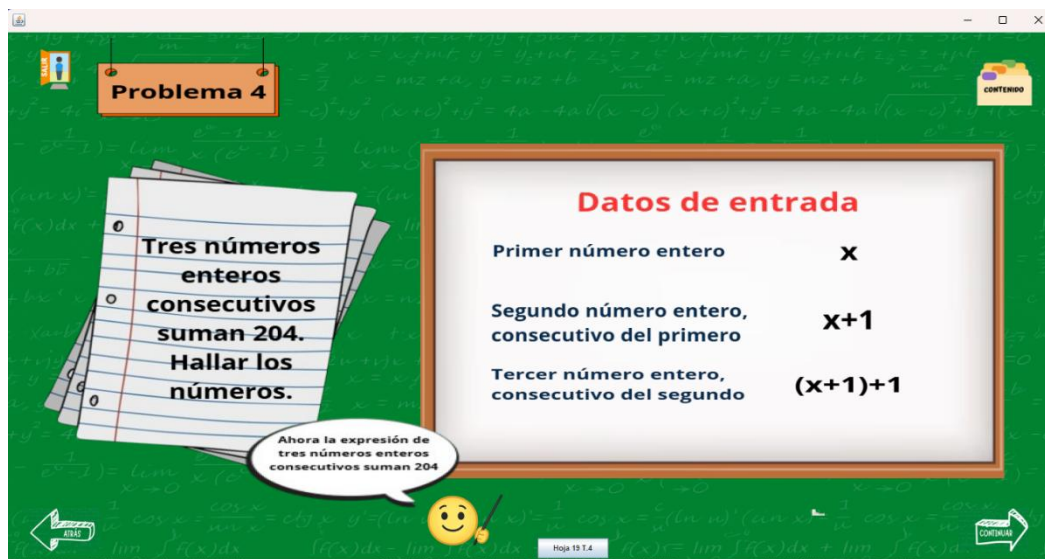


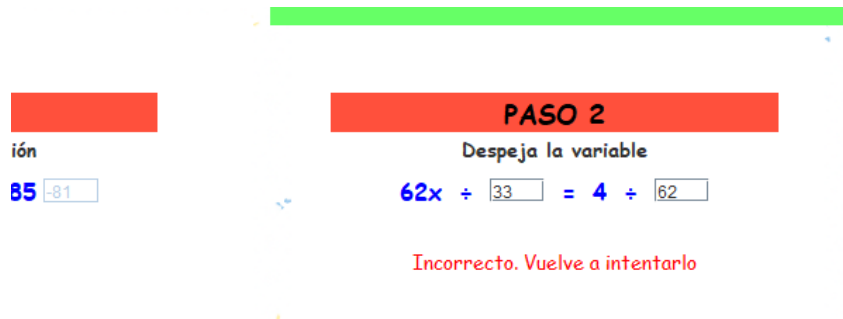
Ilustración 67 Prueba Hoja 19 Tema 4



Ilustración 68 Prueba Reto del Tema 4

## 16.4 Pruebas de manejo de errores

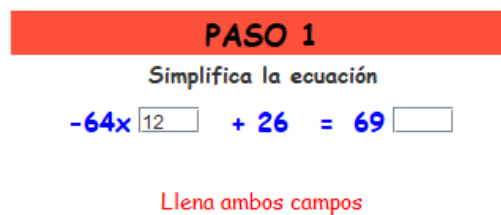
**Caso 1:** Se prueba ingresando datos incorrectos en el Paso 2 “Despeja la variable”, en donde se hace la verificación mostrando el mensaje “Incorrecto. Vuelve a Intentarlo”:



The screenshot shows a software interface for a math problem. At the top, there is a green progress bar. Below it, a red header bar contains the text "PASO 2". Underneath the header, the instruction "Despeja la variable" is displayed. The equation  $62x \div \boxed{33} = 4 \div \boxed{62}$  is shown. The input "33" in the first box is highlighted in blue. Below the equation, a red error message reads "Incorrecto. Vuelve a intentarlo". To the left of the main interface, there is a sidebar with a red header bar and the text "ión" and "85 -81" below it.

Ilustración 69 Prueba en Paso 2

**Caso 2:** Se ingresa en el Paso 1 “Simplifica la variable” datos en un solo campo, se realiza la verificación y se muestra el mensaje “Llena ambos campos”.



The screenshot shows a software interface for a math problem. At the top, there is a red header bar containing the text "PASO 1". Below the header, the instruction "Simplifica la ecuación" is displayed. The equation  $-64x \boxed{12} + 26 = 69 \boxed{\phantom{00}}$  is shown. The input "12" in the first box is highlighted in blue. Below the equation, a red error message reads "Llena ambos campos".

Ilustración 70 Prueba en Paso 1

## 16.5 Pruebas en Raspberry Pi 4

Para la ejecución del sistema necesitamos empaquetar el proyecto en un .JAR, para ello nos dirigimos al proyecto, dar clic derecho, y seleccionar Clean and Build:

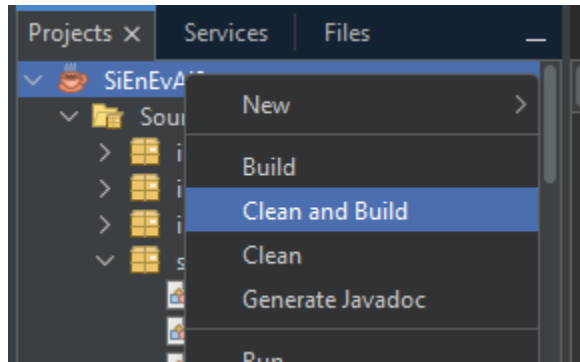


Ilustración 71 Clean and Build de Proyecto

El cual se encargará de comprimir todos los recursos del proyecto en un solo archivo, que para ubicar nos iremos a la carpeta raíz para encontrarlo.

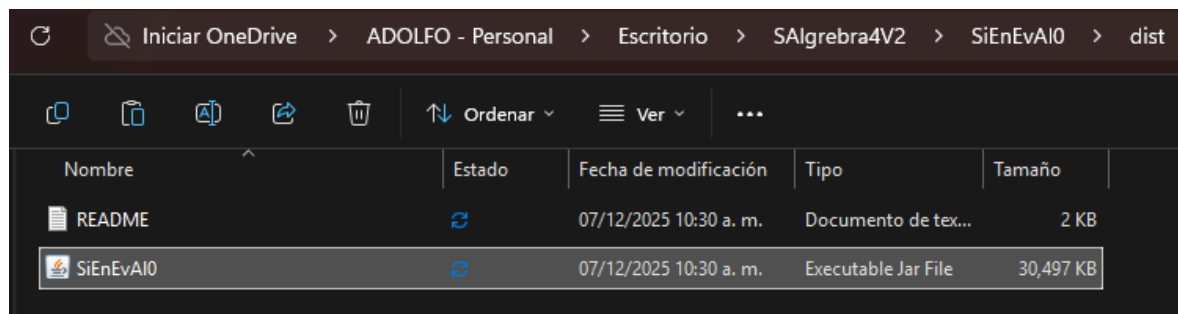


Ilustración 72 Dirección del JAR

### Pruebas de transferencia de archivos

Volviendo a WinSCP con conexión SSH, ingresamos con los datos anteriormente ingresados (Dirección IP, y credenciales de usuario), para compartir el archivo .JAR del proyecto y que este en nuestra Raspberry, por lo que una vez ingresado al menú de ingreso, solo basta ubicar la dirección del archivo, seleccionarlo y arrastrarlo al lado derecho de la pantalla, para que de manera rápida se comparta en la Raspberry, se recomienda crear una carpeta única para el archivo.

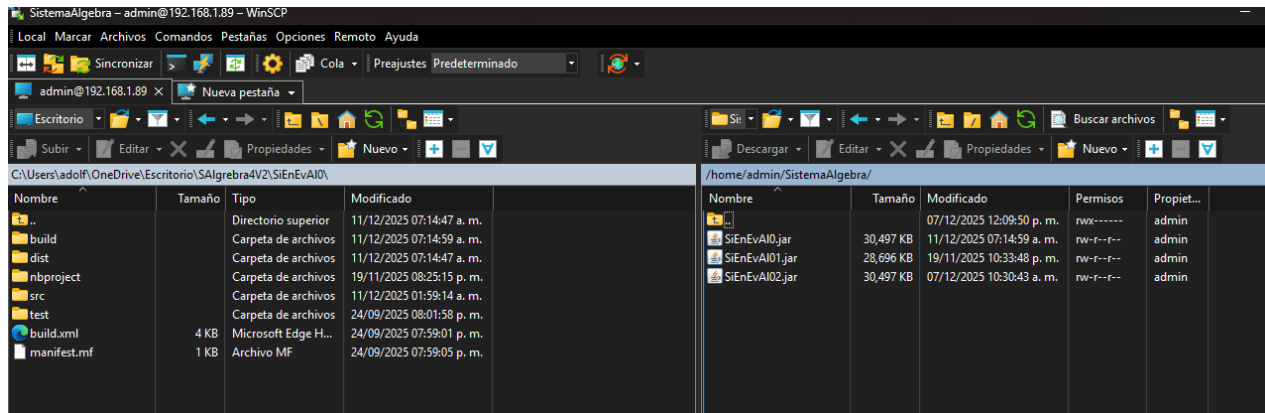


Ilustración 73 Menú de WinSCP

Por lo que una vez compartido el archivo remotamente o algún otro modo de transferencia dentro de la Raspberry Pi, podremos ejecutar archivos .JAVA o .JAR, según sea el caso.

## Caso .JAVA

En el caso de ejecutar un archivo .JAVA en la Raspberry, situados en la terminal de la Raspberry Pi y la dirección de los archivos, ejecutamos los comandos *javac nombreDelArchivo.java* para compilar y *java nombreDelArchivo* para ejecutar:

```
File Edit Tabs Help
admin@raspberrypi:~ $ cd /home/admin/5codigos
admin@raspberrypi:~/5codigos $ ls
Calculadora.java          NumeroPrimo.class      Saludo1.java
ContadorVocales.class     NumeroPrimo.java       SerieFibonacci.class
ContadorVocales.java      'Saludo1$1.class'      SerieFibonacci.java
InvertirCadena.java       Saludo1.class
admin@raspberrypi:~/5codigos $ javac Saludo1.java
admin@raspberrypi:~/5codigos $ java Saludo1
```

Ilustración 74 Comandos para un .java

Por lo que en el siguiente ejemplo se ejecutara un pequeño código con interfaz gráfica en pantalla:

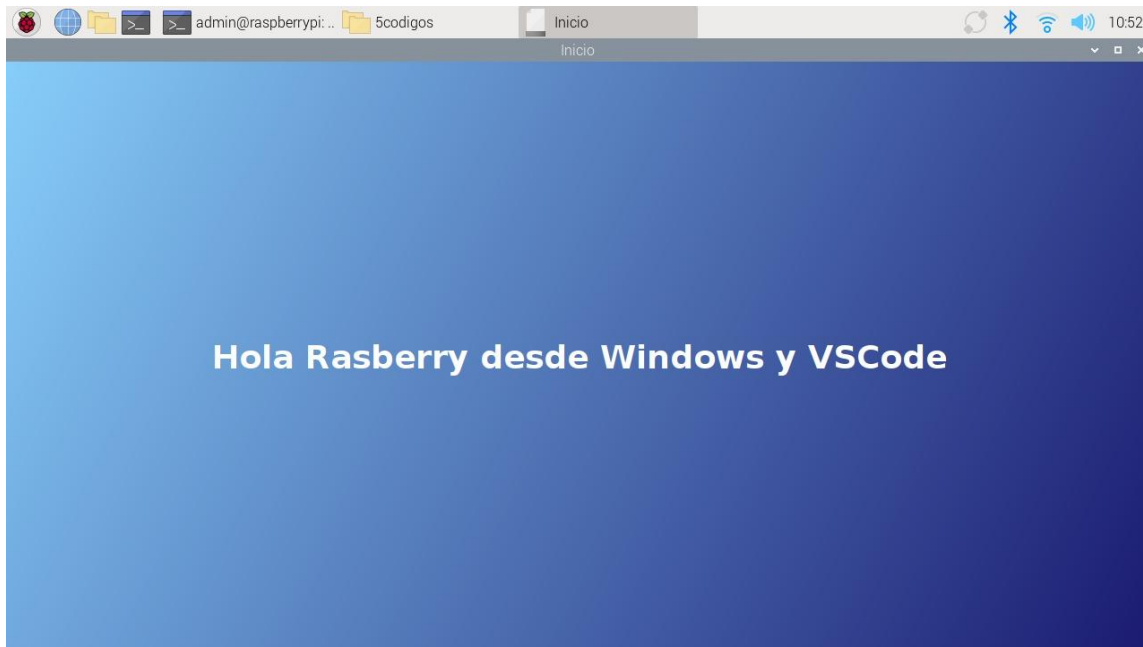


Ilustración 75 Ejecución .java

## Caso .JAR

En el caso de un .JAR (nuestro caso), solo basta con estar situado en la ruta de la dirección del archivo y ejecutar el siguiente comando *java -jar nombreDelArchivo.jar*, que ejecuta directamente el archivo:

```
admin@raspberrypi: ~/SistemaAlgebra
File Edit Tabs Help
admin@raspberrypi:~ $ cd /home/admin/SistemaAlgebra
admin@raspberrypi:~/SistemaAlgebra $ java -jar SiEnEvAl0.jar
El valor de ancho es: 1440El valor de alto es: 900El valor de a1 es 1440
El valor de a2 es 850
admin@raspberrypi:~/SistemaAlgebra $
```

Ilustración 76 Comandos para .jar

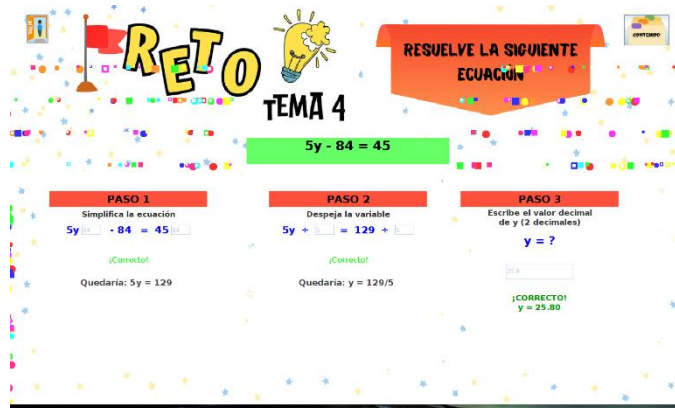
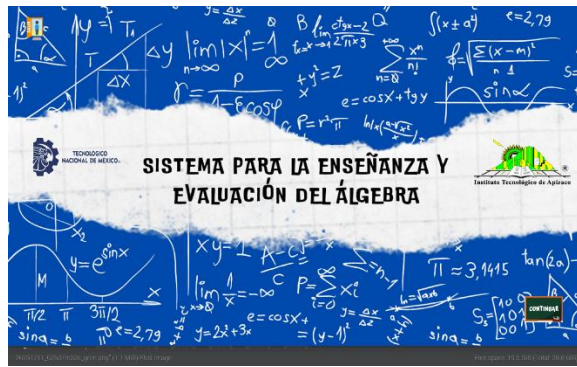


Ilustración 77 Pruebas de Interfaz en Raspberry PI OS

## Resultados de pruebas en Raspberry Pi:

- **Arranque exitoso:** Desde comando hasta interfaz visible.
- **Rendimiento estable:** Animaciones estables.
- **Consumo de memoria estable.**
- **Visualización correcta:** La responsividad se mantiene.

## 17. Pruebas de usabilidad

### 17.1 Metodología de pruebas de usabilidad

Se realizaron pruebas con 3 perfiles de usuario:

1. **Estudiante principiante** (Segundo año de secundaria).
2. **Estudiante avanzado** (Primer semestre de ingeniería).
3. **Docente de matemáticas** (experto en el tema).

### 17.2 Escenarios de prueba

#### Escenario 1: Primer uso del sistema

**Objetivo:** Evaluar curva de aprendizaje.

**Tareas:**

1. Iniciar aplicación.
2. Resolver primera ecuación guiada.
3. Completar los 3 pasos.
4. Ver animación de confeti.

Métrica: Tiempo hasta completar primera ecuación: 2-3 minutos.

#### Escenario 2: Uso repetido

**Objetivo:** Evaluar eficiencia.

**Tareas:**

1. Resolver 5 ecuaciones consecutivas.
2. Usar atajos de teclado (Enter para validar).
3. Navegar entre pasos sin dificultad.

Métrica: Tiempo por ecuación en intentos 60 segundos – 2 minutos.

### **Escenario 3: Recuperación de errores**

Objetivo: Evaluar robustez.

Tareas:

1. Introducir respuesta incorrecta.
2. Leer mensaje de error.
3. Corregir y continuar.

Métrica: Todos los usuarios corrigieron en <5 intentos.

### **17.3 Criterios de usabilidad evaluados**

#### **1. Facilidad de aprendizaje:**

- Todos los usuarios completaron primera ecuación sin ayuda externa.
- Instrucciones claras en cada paso.
- Retroalimentación inmediata.

#### **2. Eficiencia de uso:**

- Atajo Enter para validación rápida.
- Progreso visual claro.

#### **3. Memorabilidad:**

- Usuarios recordaron flujo en segunda sesión.
- Patrones consistentes en los 3 pasos.

#### **4. Satisfacción subjetiva:**

- Comentarios positivos sobre animaciones y confeti.
- Sugerencia: Añadir sonido de celebración y animaciones extras.

#### **5. Prevención de errores:**

- Validación en tiempo real.
- Mensajes de error específicos.

# CONCLUSIONES

## Conclusiones

El desarrollo e implementación de interfaces gráficas responsivas para el módulo de evaluación del tema de resolución de ecuaciones lineales de un tutor cognitivo para álgebra dentro del sistema educativo de álgebra para el Departamento de Posgrado e Investigación del Instituto Tecnológico de Apizaco culminó exitosamente, logrando los siguientes hitos significativos:

La Implementación técnica se completó, ya que se desarrolló un módulo educativo funcional que incluye interfaces gráficas responsivas, un algoritmo generador de ecuaciones lineales adaptado, un sistema paso a paso integrado y un sistema de animaciones.

La integración del módulo al sistema existente se incorporó coherentemente a la arquitectura del software educativo preexistente, manteniendo consistencia visual y funcional con los demás temas, demostrando capacidad de trabajo en entornos de software establecidos.

El sistema fue probado y validado exitosamente en diferentes entornos, incluyendo computadoras con Windows y la plataforma embebida Raspberry Pi 4, comprobando su portabilidad y responsividad en hardware diverso.

Se implementó un enfoque de enseñanza único mediante un sistema tutorial interactivo de tres pasos que guía al estudiante en el proceso de resolución de ecuaciones lineales, combinando instrucción estructurada con práctica inmediata y retroalimentación personalizada.

## Recomendaciones

Basado en el análisis del sistema desarrollado y considerando oportunidades de mejora identificadas durante el proceso de desarrollo e implementación, propondría las siguientes recomendaciones

- **Algoritmo de dificultad:** Implementar un algoritmo que analice el desempeño del estudiante y ajuste automáticamente la dificultad de las ecuaciones:
  - **Principiante:** Coeficientes entre 1-20, siempre positivos, resultados enteros
  - **Intermedio:** Coeficientes 1-50, incluye negativos, algunos resultados decimales simples
  - **Avanzado:** Coeficientes 1-99, mezcla de positivos/negativos, resultados decimales complejos
- **Logros y medallas:** Implementar un sistema de recompensas por avances específicos.
- **Desbloqueo de niveles:** Desbloquear temas o modos especiales al alcanzar cierta cantidad de puntos.
- **Voz de guía:** implementar voz o sonido que guíe al usuario a través del contenido teórico

## **Experiencia adquirida**

Durante estas residencias transformé mi conocimiento teórico de Java y desarrollo de software en habilidades prácticas aplicadas, pasé de crear ejercicios académicos a desarrollar un sistema educativo completo que integra algoritmos matemáticos, interfaces gráficas responsivas, animaciones pedagógicas y validación en tiempo real. Por lo que aprendí que el desarrollo profesional no es solo escribir código, sino resolver problemas reales considerando usuarios, hardware limitado y objetivos pedagógicos.

## **Aprendizaje Personal**

En este proyecto aprendí haciéndolo: descubrí cómo realmente funcionan mucha de la sintaxis en el manejo de Java, como layouts responsivos, manejar recursos de imágenes eficientemente, y cómo implementar sistemas complejos como animaciones y validaciones matemáticas. También aprendí perseverancia técnica: cuando los cálculos de responsividad no funcionaban en la Raspberry Pi o las interfaces se veían mal en ciertas resoluciones o animaciones, no me rendí, sino que aprendí a depurar y optimizar hasta que funcionó.

## **Crecimiento Profesional**

Profesionalmente, pasé de seguir instrucciones a tomar decisiones técnicas informadas, ya que aprendí a analizar código heredado (como el algoritmo con exponentes y la estructura de las interfaces preexistentes) y mejorarlo según necesidades reales. Trabajar con mi asesora, la Dra. Blanca Estela, fue fundamental: porque me enseñó a ver más allá del código, a entender las necesidades pedagógicas reales detrás de cada función, y a comunicar soluciones técnicas en términos de beneficios educativos. Y lo más importante: entendí que, en el desarrollo profesional, "funciona" no es suficiente - debe funcionar bien, en diferentes dispositivos, para diferentes usuarios, cumpliendo objetivos reales más allá de lo técnico.

## Competencias desarrolladas

Durante el desarrollo del proyecto aprendí y fortalecí mis habilidades en las siguientes competencias:

**Desarrollo de Software en Java:** Programar interfaces gráficas responsivas usando layouts proporcionales y cálculos, además de hacer uso de JTextPane con HTML, JLabels animados, sistemas de eventos complejos fortaleció mis conocimientos en Java.

**Optimización de recursos gráficos:** Procesamiento de imágenes PNG transparentes, escalado eficiente, carga diferida, es una forma de trabajo más limpio y por ende se ve mucho más profesional.

**Pruebas y Calidad de Software:** Realizar Pruebas de usabilidad con usuarios reales con estudiantes y docentes evaluando flujo pedagógico, la validación multiplataforma por medio del testing en diferentes resoluciones y hardware.

**Gestión de Proyectos de Software:** Documentación técnica y código comentado, y reportes de avance.

**Cumplimiento de cronograma:** Entrega de hitos en tiempos establecidos

## Referencias

Brendan, B. A. (2016). Technology usage in mathematics education research – A systematic review of recent trends. *Computers & Education*, 255-273.

Canva. (2025). *Home*. Canva. Retrieved from <https://www.canva.com/>

Corcuera, P. (2025). *Interfaces de usuario con AWT y Swing*. Universidad de Cantabria. Retrieved from [https://personales.unican.es/corcuerp/java/Slides/GUIs\\_AWT\\_Swing.pdf](https://personales.unican.es/corcuerp/java/Slides/GUIs_AWT_Swing.pdf)

Cuemath.com. (2025). *Linear equations*. Retrieved from <https://www.cuemath.com/algebra/linear-equations/>

Design, G. M. (2023). *Animation: Meaningful transitions*. Retrieved from <https://m3.material.io/styles/motion/overview>

Donald, N. (2013). *The Design of Everyday Things*. New York: Basic Books.

J., S. C. (2013). *Computer Aided Assessment of Mathematics*. Oxford: Oxford University Press.

Jiménez-Hernández, E. M. (2012). *Metodología Híbrida para Desarrollo de Software en México*. Tecnología de Software y Bases de Datos, Centro de Investigación en Computación (CIC), IPN.

Lenguajecss. (2025). *Funciones exponenciales CSS*. Lenguajecss.com. Retrieved from <https://lenguajecss.com/css/funciones-css/funciones-exponenciales/>

M, D. (2020). *La Guía para Principiantes del Diseño Web Responsivo*. Kinsta. Retrieved from <https://kinsta.com/es/blog/disenio-de-paginas-web-sensibles/>

Mayer, R. E. (2002). *Animation as an Aid to Multimedia Learning*. Educational Psychology Review. Retrieved from <https://doi.org/10.1023/A:1013184611077>

MSMK. (2025). *¿Qué es AWT (Abstract Window Toolkit)?* MSMK. Retrieved from <https://msmk.university/que-es-awt-abstract-window-toolkit/>

NetBeans, A. (2025). *Apache NetBeans releases*. Apache.org. Retrieved from <https://netbeans.apache.org/front/main/download/>

NetBeans, A. (2025). *Introduction to GUI Building*. Apache.org. Retrieved from <https://netbeans.apache.org/tutorial/main/kb/docs/java/gui-functionality/>

Oracle. (2025). *Java downloads*. Oracle.com. Retrieved from <https://www.oracle.com/java/technologies/downloads/>

Panella. (2023). *Gamification: qué es la Gamificación y cómo funciona*. Thinking for Innovation. Retrieved from <https://www.iebschool.com/hub/gamification-innovacion/>

Pi, R. (2019). *¿Que es Raspberry Pi?* Raspberry Pi; MCI Electronics. Retrieved from <https://raspberrypi.cl/que-es-raspberry/>

Pi, R. (2025). *Downloads*. Raspberrypi.com. Retrieved from <https://www.raspberrypi.com/software/>

PuTTY, D. (2025). *a free SSH and telnet client for Windows*. Putty.org. Retrieved from <https://putty.org/index.html>

Specialist., E.-L. (2021). *Importancia de las pizarras digitales interactivas en aulas virtuales*. CAE Computer Aided E-learning. Retrieved from <https://www.cae.net/es/importancia-pizarras-digitales-interactivas-aulas-virtuales/>

Uxcel. (2025). *Where modern product professionals learn*. Retrieved from <https://app.uxcel.com/glossary/responsive-design>

Valtx. (2022). *Metodologías de desarrollo de software: ¿Qué son y para qué sirven?* Valtx. Retrieved from <https://www.valtx.pe/blog/metodologias-para-el-desarrollo-de-software-que-son-y-para-que-sirven>

WinSCP. (2025). *Official site :: Download*. Winscp.net. Retrieved from <https://winscp.net/eng/download.php>