

**Ecuplot WebApp: Visualización Interactiva de Funciones Matemáticas con Autenticación  
e Historial de Gráficas para Estudiantes y Docentes**

Julian Alejandro Cardenas Santiago, Carlos Mateo Cruz Ibarra

Universidad de Cundinamarca, Facultad de Ingeniería

Ingeniería de Sistemas y Computación

Ingeniería de Software 2

Ing. German Ortiz Diaz

8 de octubre del 2025

## Resumen

Ecuplot WebApp nace de migrar un bot de Telegram hacia una aplicación web orientada a estudiantes y docentes STEM. El objetivo es facilitar la visualización interactiva de funciones matemáticas (algebraicas, trigonométricas y más) y conservar un historial por usuario para retomar ejercicios y comparar resultados. La solución integra dos modos de uso (invitado y registrado), **autenticación con verificación por correo y 2FA**, y un modulo de **graficas interactivas** que mejora la experiencia frente a imágenes estáticas al permitir zoom, paneo y lectura puntual de valores. El enfoque responde a barreras frecuentes en contextos educativos reales —tiempo de clase limitado, heterogeneidad de dispositivos y conectividad variable—, priorizando una interfaz ligera y clara. Además, el uso de librerías de visualización web modernas permite prototipar y escalar sin sacrificar usabilidad. En suma, EcuPlot busca reducir fricción entre teoría y práctica: que el estudiante “vea” la matemática, y que el docente recupere intentos, anote patrones y retroalimente mejor.

**Palabras clave:** Autenticación, EcuPlot, educación STEM, Telegram, web.

## Abstract

EcuPlot WebApp was born from migrating a Telegram bot into a web application aimed at STEM students and teachers. The objective is to facilitate the interactive visualization of mathematical functions (algebraic, trigonometric, and more) and maintain a history per user to resume exercises and compare results. The solution integrates two usage modes (guest and registered), authentication with email verification and 2FA, and an interactive graphics module that improves the experience compared to static images by allowing zooming, panning, and reading values on-demand. The approach addresses common barriers in real-world educational contexts—limited class time, device heterogeneity, and variable connectivity—by prioritizing a lightweight and clear interface. Furthermore, the use of modern web visualization libraries allows for prototyping and scaling without sacrificing usability. In short, EcuPlot seeks to reduce friction between theory and practice: allowing students to "see" the mathematics, and allowing teachers to recover attempts, note patterns, and provide better feedback.

**Keywords:** Authentication, EcuPlot, STEM education, Telegram, web.

## Contenido

Resumen.....	2
Abstract .....	3
Figuras .....	9
Introducción.....	10
Antecedentes .....	12
Marco Referencia.....	13
Marco Teórico.....	14
Visualización e Interactividad .....	14
Aprendizaje Matemático y Habilidades Espaciales .....	15
Acceso y Diseño Sensible a la Conectividad .....	15
Seguridad Aplicada a Educación .....	15
Marco de Antecedentes .....	15
Marco Conceptual .....	17
Aplicación Educativa .....	17
Función Matemática .....	17
Representación Gráfica .....	17
Bot de Telegram.....	17
API.....	17
Python .....	17
Visual Studio Code .....	18
Base de Datos (historial) .....	18
Aprendizaje Interactivo .....	18

Planteamiento del Problema .....	19
Justificación.....	20
Objetivos .....	22
Objetivo General.....	22
Objetivos específicos.....	22
Metodología .....	24
Fases del proyecto bajo XP .....	24
Requerimientos .....	25
Requerimientos Funcionales .....	25
Requisitos no Funcionales .....	26
Casos de Uso.....	27
Actores .....	27
Catalogo de Casos de Uso.....	28
Pruebas Unitarias.....	34
Análisis de resultados.....	35
Historias de Usuario.....	37
Product Backlog .....	38
Funcionalidades .....	39
Diagramas.....	42
Casos de Uso .....	42
Secuencia.....	43
Actividades .....	45
Colaboración .....	46

Máquina de Estado.....	47
Componentes .....	48
Despliegue.....	48
Diagrama Entidad Relación .....	49
Metodología XP aplicada al proyecto.....	52
Principios adoptados .....	52
Roles del Equipo .....	52
Practicas XP .....	52
Plan de Iteraciones (2 semanas c/u) .....	52
Actas de Reunion.....	53
Acta de Reunión XP No. 1.....	53
Decisiones tomadas:.....	53
Acta de Reunión XP No. 2.....	54
Decisiones tomadas:.....	54
Acta de Reunión XP No. 3.....	54
Temas tratados: .....	55
Decisiones tomadas:.....	55
Acta de Reunión XP No. 4.....	55
Temas tratados: .....	55
Decisiones tomadas:.....	56
Acta de Reunión XP No. 5.....	56
Temas tratados: .....	56

Decisiones tomadas:.....	56
Acta de Reunión XP No. 6.....	57
Temas tratados: .....	57
Decisiones tomadas:.....	57
Acta de Reunión XP No. 7.....	58
Temas tratados: .....	58
Decisiones tomadas:.....	58
Historial de Reuniones Semanales .....	59
Impacto del Proyecto .....	60
Impacto en el Aprendizaje y en el Estudiante .....	60
Impacto Institucional y de Gestión del Conocimiento .....	61
Impacto Social y en Equidad .....	61
Impacto Esperado Específico de EcuPlotWeb .....	62
Aporte de los Otros Cadi del Semestre.....	62
Ingeniería de Software II .....	63
Networking.....	63
Comunicación de Datos .....	63
Conclusiones.....	65
La visualización si acerca la matemática .....	65
La Arquitectura Quedó Lista para Crecer sin Romperse .....	65
Las Piedras del Camino Están Claras .....	65
Lo que Sigue es Validar con Aula Real.....	65

Anexos .....	66
Enlace Github .....	66
Enlace Airtable.....	66
Referencias .....	67

## Figuras

Figura 1 .....	¡Error! Marcador no definido.
Figura 2 .....	¡Error! Marcador no definido.
Figura 3 .....	¡Error! Marcador no definido.
Figura 4 .....	50
Figura 5 .....	50
Figura 6 .....	51
Figura 7 .....	51

## Introducción

EcuPlot WebApp surge como la evolución de un bot de Telegram hacia una aplicación web centrada en el **aprendizaje de matemáticas con graficas interactivas**. La motivación es práctica: en clase el tiempo es corto, los dispositivos de los estudiantes varía y la conectividad no siempre acompaña; por ello se requiere una herramienta ligera, clara y capaz de **poner la función “en la mano”** con zoom, paneo y lectura puntual de valores. La literatura respalda esta apuesta: las **visualizaciones** y, en particular, las **interacciones dinámicas** muestran efectos **positivos y de magnitud media** en el aprendizaje matemático, facilitando la comprensión de conceptos abstractos y el razonamiento espacial.

En términos técnicos, la solución adopta librerías de visualización que soportan interacciones nativas (hover, selección y reacomodo del lienzo), lo que **reduce fricción** frente a imágenes estáticas y habilita análisis más fino por parte del usuario. Estas capacidades permiten que el gráfico funcione **como entrada y como salida** de la experiencia: el usuario explota y, al explorar, produce datos útiles para retroalimentación docente.

El proyecto incorpora **autenticación** con verificación por correo y **2FA**, cuidando seguridad y privacidad sin sacrificar usabilidad. Las guías de OWASP recomiendan **MFA/2FA** como control crítico contra la mayoría de los ataques relacionados con contraseñas, por lo que se incluye como estándar de calidad desde el diseño.

Contextualmente, EcuPlot busca ser útil en escenarios de **conectividad desigual**, comunes en zonas rurales y periurbanas; en Colombia, el Ministerio TIC ha priorizado iniciativas para cerrar brechas mediante despliegue de infraestructura y soluciones de acceso comunitario, subrayando la necesidad de herramientas educativas web **eficientes y adaptables**.

En suma, la contribución es doble: **pedagógica**, al acercar la matemática a través de interacciones que promueven exploración guiada, y **tecnológica**, al ofrecer la arquitectura web modular con historial por usuario, criterios de accesibilidad y seguridad fortalecida. El resto del

documento presenta requerimientos, casos de uso, historias de usuario, diseño (componentes, datos y APIs), pruebas y resultados, así como la planificación ágil y la alineación curricular que enmarcan el desarrollo de EcuPlot.

## Antecedentes

EcuPlot Web nace de una experiencia previa: un bot de Telegram que recibía funciones y devolvía una imagen con su respectiva grafica. Lo que funciono como “primer puente” entre el estudiante y la visualización, hoy pide algo más: **interacción directa** (acercar, mover, leer valores) y **memoria de trabajo** (historial por usuario para retomar intentos y comparar). Esta evolución también dialoga con una realidad cotidiana: el **tiempo de clase es corto**, los dispositivos cambian entre estudiantes y la **conectividad no siempre acompaña**. En Colombia, por ejemplo, el propio ministerio reporta avances fuertes en accesos, pero con brechas persistentes entre territorios y tipos de conexión (MinTIC, 2024). A nivel global, los informes de la UIT recuerdan que el desafío no es solo “llegar” a internet, sino **llegar con calidad significativa**; en zonas rurales todavía predominan coberturas móviles de menor desempeño y una “brecha de uso” amplia frente a la oferta, lo cual impacta el aprendizaje en línea y las apps educativas (ITU, 2022a, 2022b).

En paralelo, la investigación educativa ha acumulado evidencia: **aprender con visualizaciones ayuda**. Meta-análisis y revisiones recientes muestran efectos positivos—de magnitud media—in la comprensión de conceptos matemáticos al introducir **visualizaciones externas y tareas espaciales** que el estudiante pueda manipular (Schoenherr, 2024; Lowrie et al., 2023). Dicho simple: cuando el grafico se deja tocas, **las ideas se vuelven más visibles** y menos abstractas.

Desde lo técnico, hoy contamos con librerías que no solo “dibujan” sino que escuchan al usuario: el grafico puede ser **entrada y salida** a la vez (por ejemplo, captar un clic, un área seleccionada o un cambio de escala para responder con nuevos datos). Esto permite construir experiencias más ricas sin exigir instalaciones pesadas, solo navegador (Plotly; Dash).

Por último, la seguridad dejó de ser un extra. Si vamos a guardar historiales y cuentas, debemos proteger el acceso. Las guías de OWASP recomienda **MFA/2FA** como una defensa

clave contra la mayoría de los ataques ligados a contraseñas, por eso el proyecto la incorpora desde el diseño (OWASP, 2024a, 2024b).

### **Marco Referencia**

EcuPlot nace en la intersección entre tres mundos que se tocan cada vez más: la enseñanza de matemáticas con apoyo visual, la programación en Python y la mensajería como interfaz de usuario. La apuesta es sencilla de decir y difícil de ejecutar: que el estudiante escriba una función como la pensaría en el cuaderno y reciba la gráfica lista, sin pelear con software complejo ni licencias.

Por el lado del **lenguaje y las librerías**, Python se volvió el “destornillador” estándar de la caja: su sintaxis amable y la comunidad enorme hacen que piezas como *NumPy*, *SymPy* y *Matplotlib* encajen sin mucha fricción. En la práctica, *Matplotlib* es el caballo de batalla para renderizar la gráfica con ejes limpios y rangos controlados; es estable, documentado y funciona bien tanto en local como en servidores livianos (*Matplotlib documentation*, n.d.). Esa base técnica es la que permite que la experiencia sea inmediata: el usuario envía la función, el motor calcula y la imagen vuelve al chat.

En cuanto a la **puerta de entrada**, Telegram es menos un “chat” y más un sistema operativo diminuto. Sus *bots* no solo reciben comandos; pueden validar entradas, generar archivos y mantener un hilo de conversación útil. La **Telegram Bot API** está bien documentada y tiene soporte sólido para enviar imágenes y manejar errores con mensajes claros, justo lo que EcuPlot necesita para no romper el flujo del estudiante (*Telegram Bot API*, n.d.). Para el desarrollo, **Pyrogram** facilita el trabajo asíncrono y la estructura por *plugins*, que encaja con la idea de ir sumando funciones sin desordenar el proyecto (*Welcome to Pyrogram*, n.d.; TeLe TiPs, 2021).

Desde **lo pedagógico**, lo que EcuPlot pone en juego no es solo “hacer más bonita la matemática”, sino permitir ese ciclo de prueba-error rápido que el papel no da: cambiar un exponente, ver cómo se abre la parábola; sumar un seno y notar la ondulación sobre una recta.

Esa ida y vuelta concreta ayuda a fijar conceptos abstractos y da pie a preguntas mejores (“¿qué pasa si acoto el dominio?”, “¿y si compongo funciones?”). Este tipo de andamiaje visual favorece el aprendizaje activo y la conexión entre representación simbólica y geométrica (Ausubel, 2002; Cabero & Llorente, 2020).

Finalmente, **la persistencia** importa: guardar lo que el usuario probó (función, fecha, usuario) no es un “extra”, es memoria de trabajo para retomar procesos, comparar variantes y, con el tiempo, detectar patrones personales de aprendizaje. Ese registro simple —bien diseñado y respetando privacidad— convierte una conversación cualquiera en un rastro de estudio útil.

En conjunto, estas referencias arman el mapa: Python y sus librerías para el “cómo”, Telegram/Pyrogram para el “por dónde”, y la didáctica visual para el “para qué”. EcuPlot se para ahí: rápido, claro y lo bastante flexible para crecer con nuevas funciones sin romper la experiencia (Pro-Ciencia, 2018; Matplotlib documentation, n.d.; Telegram Bot API, n.d.; Welcome to Pyrogram, n.d.).

### **Marco Teórico**

#### **Visualización e Interactividad**

Llamamos “visualización externa” a los apoyos (gráficos, diagramas, animaciones) que se sitúan fuera de la mente del estudiante y que organizan la información para pensar con ella. La literatura reciente caracteriza este campo como diverso, pero consistente en un punto: **hace visible lo invisible** reduce carga cognitiva innecesaria y favorece el razonamiento matemático (Schoenherr, 2024). Cuando esa visualización **responde a la acción** (acercar, señalar, seleccionar), cambia el papel del estudiante: deja de “mirar un resultado” y pasa a **explorar un fenómeno**.

En aplicaciones web modernas, el componente gráfico puede emitir eventos (hover, clic, selección) y detonar actualizaciones en tiempo real: filtrar, recalcular, anotar. Este ida y vuelta con el usuario convierte la gráfica en un espacio de diálogo con los datos (Dash: Plotly). Para

un curso de matemáticas, ese diálogo se traduce en gestos sencillos —mover la vista, leer coordenadas, comparar curvas—que apoyan explicaciones del docente y descubrimientos del estudiante.

### **Aprendizaje Matemático y Habilidades Espaciales**

Intervenciones que entrenan **visualización y rotación mental** han mostrado mejoras en desempeño matemático, en especial en geometría y medición; los efectos no son mágicos, pero si **prácticos y sostenidos** cuando se integran a la enseñanza regular (Liwrie et al., 2023). EcuPlot se apoya en este marco: no busca reemplazar la clase, sino **darle herramientas** para que el estudiante pruebe, compare y pregunte mejor.

### **Acceso y Diseño Sensible a la Conectividad**

En contextos con **internet inestable o móvil** (comunes en áreas rurales), conviene privilegiar **cargas ligeras**, renderizado en el cliente y **progressive enhancement**: que lo esencial funcione con recursos mínimos y que las funciones avanzadas se activen cuando haya ancho de banda suficiente. Los reportes globales y locales sobre conectividad hacen explícito este reto: no basta con “estar en línea”, la **calidad de la conexión** condiciona el uso educativo real (ITU, 2022<sup>a</sup>, 2022b; MinTIC, 2024).

### **Seguridad Aplicada a Educación**

Si guardamos **históricos por usuario** (intentos, funciones, fechas0), debemos proteger credenciales y sesiones. OWASP sugiere **MFA/2FA** cuando sea viable para la población, como barrera efectiva ante ataques de fuerza bruta y robos de contraseña; además, recomienda prácticas básicas (hash seguro, rotación de tokens, límites de intentos), que EcuPlot adopta en su arquitectura (OWASP, 2024a, 2024b).

### **Marco de Antecedentes**

La idea de **EcuPlot** surgió como una respuesta a un problema cotidiano en cursos STEM: la necesidad de **visualizar funciones de manera inmediata, clara y reproducible**. El primer experimento fue un **bot de Telegram** escrito en **Python** que recibía una ecuación en

texto, validaba su sintaxis y devolvía una **imagen** con la gráfica generada. Esta prueba de concepto capitalizó bibliotecas consolidadas del ecosistema científico de Python—particularmente **Matplotlib** para el renderizado de curvas—y un **wrapper** ágil para la API de Telegram como **Pyrogram**, que simplificó la comunicación cliente-bot y el manejo de mensajes (Matplotlib, s. f.; Pyrogram, s. f.; Telegram Bot API, s. f.).

Durante esa fase inicial se confirmaron tres hallazgos: (1) la **factibilidad técnica** de interpretar expresiones algebraicas y trigonométricas y graficarlas con tiempos de respuesta bajos; (2) el **interés real de usuarios** (compañeros y docentes) por una herramienta ligera, sin fricción de instalación; y (3) ciertas **limitaciones** del canal de mensajería—por ejemplo, la dependencia de conectividad permanente y restricciones de presentación visual dentro del chat. En la práctica, tutoriales y guías técnicas disponibles públicamente aceleraron la curva de aprendizaje para el armado del bot y el pipeline de graficación (Pro-Ciencia, 2018; TeLe TiPs, 2021).

A partir de esos aprendizajes, el proyecto evolucionó hacia una **aplicación educativa independiente** con mayor control sobre la interfaz, la persistencia y el historial de consultas. Se mantuvo Python como núcleo por su **ecosistema maduro** y por la continuidad con el prototipo; asimismo, se adoptó una **base de datos relacional** para registrar funciones, usuarios y metadatos mínimos, con el fin de **recuperar procesos previos** y dar seguimiento a sesiones de estudio. El entorno de desarrollo se consolidó en **Visual Studio Code**, que permitió estandarizar extensiones, linters y flujos de prueba, y dejó abierta la puerta para módulos de seguridad y escalabilidad más robustos en la versión web (Matplotlib, s. f.; Pyrogram, s. f.).

En términos de trayectoria, EcuPlot ha seguido un ciclo iterativo claro: **idea → prototipo funcional (bot) → validación de uso → identificación de cuellos de botella → rediseño hacia aplicación web**. Cada iteración depuró el objetivo original—**aprender con gráficos**—y lo aterrizó en decisiones concretas: separación de responsabilidades (parsing, evaluación,

render), manejo de errores con mensajes comprensibles, y almacenamiento mínimo pero útil para la continuidad del aprendizaje. Este recorrido proporciona el **piso técnico y pedagógico** para la fase actual, enfocada en una experiencia de usuario más rica y en la sostenibilidad del proyecto dentro de contextos académicos.

### **Marco Conceptual**

#### **Aplicación Educativa**

Software pensado para aprender haciendo. En EcuPlot, la app no da clases: abre atajos. Recibe una función, devuelve una gráfica limpia y deja al estudiante explorar sin miedo a “dañar” nada.

#### **Función Matemática**

Regla que empareja valores de entrada con salidas. Aquí, la escribes en el chat y el sistema la evalúa en un rango seguro para dibujar su comportamiento de forma legible.

#### **Representación Gráfica**

La foto de la idea. Ver la curva —y no solo leer la ecuación— revela simetrías, crecimientos, cortes con ejes y efectos de cambiar un parámetro. Es la pista rápida entre la teoría y la intuición.

#### **Bot de Telegram**

Asistente dentro del chat. Recibe mensajes, valida la expresión, llama al motor de cálculo y responde con la imagen. Si algo falla, avisa y sugiere cómo corregir, sin sacar al usuario del flujo.

#### **API**

Puente de comunicación entre piezas. EcuPlot usa la **Telegram Bot API** y **Pyrogram** para orquestar mensajes, comandos y archivos con el servidor que hace los cálculos.

#### **Python**

El motor bajo el capó. Combina legibilidad con librerías científicas maduras para evaluar expresiones y graficar con precisión.

### **Visual Studio Code**

El taller. Permite desarrollar, depurar y organizar el proyecto en módulos y *plugins* sin perderse en configuraciones.

### **Base de Datos (historial)**

La memoria del proceso. Guarda qué se graficó y cuándo, para retomar caminos, comparar variaciones y aprender de lo ya hecho.

### **Aprendizaje Interactivo**

Modelo donde el estudiante mueve perillas y ve consecuencias al instante. En EcuPlot, cada ajuste en la función es una nueva imagen: menos teoría en abstracto, más “lo veo y lo entiendo” (Cabero & Llorente, 2020).

## Planteamiento del Problema

En el proceso de formación de los estudiantes de Ingeniería de Sistemas y Computación, es común encontrarse con un reto que parece simple, pero que termina siendo decisivo: lograr que una función matemática escrita en papel cobre vida en una gráfica clara y comprensible. En la práctica, muchos jóvenes llegan a sentir que ese “puente” entre lo algebraico y lo visual es difícil de cruzar, lo que repercute en su comprensión y en la confianza para resolver problemas.

Investigaciones en contextos universitarios muestran que esta transición es una de las dificultades más persistentes en matemáticas, incluso en funciones básicas como las lineales y cuadráticas (Castro-Linares & Peña-Borrero, 2019).

La literatura coincide en que las representaciones gráficas no son solo un apoyo estético, sino que potencian de forma real la comprensión conceptual.

Cuando los estudiantes pueden visualizar lo que antes era solo una ecuación, la motivación y el rendimiento tienden a mejorar, y los aprendizajes permanecen más tiempo (Li et al., 2019). Sin embargo, en la vida académica diaria, no siempre se cuenta con herramientas que hagan ese proceso sencillo.

El problema es que muchas de las aplicaciones para graficar funciones son de pago, complejas de instalar o demandan un conocimiento técnico previo que termina desmotivando a quienes solo buscan un apoyo rápido para estudiar. Además, en América Latina se han identificado barreras como limitaciones de conectividad, falta de licencias institucionales y poca formación docente en el uso de herramientas digitales, lo que restringe su adopción en programas universitarios (Delgado & Sánchez, 2018). Este panorama termina dejando a estudiantes y profesores sin un recurso inmediato y accesible.

Aquí nace la idea de EcuPlot Web, una aplicación pensada como un sistema de chat sencillos y gratuito. El estudiante o docente podrá escribir la función y recibir de inmediato la gráfica correspondiente. Quien desee llevar un seguimiento tendrá la opción de registrarse,

iniciar sesión y guardar su historial; y quien solo necesite una consulta rápida podrá entrar como invitado sin registrarse. Este enfoque busca responder a una necesidad práctica: contar con un apoyo académico sin fricciones, fácil de acceder desde la web y disponible para todos.

El valor de un proyecto así no se limita al beneficio individual. Cuando la información se organiza y se conserva, se convierte en conocimiento útil para el colectivo. En el campo educativo, la gestión del conocimiento implica justamente rescatar experiencias, compartirlas y hacerlas parte de un proceso continuo de mejora (Chamorro-Estupiñán et al., 2023). Sin un sistema que facilite este tránsito, los aprendizajes se fragmentan, se pierde continuidad entre cursos y los proyectos se ven obligados a empezar de cero una y otra vez.

Por último, vale mencionar que los sistemas conversacionales en educación han mostrado un potencial importante para acompañar y motivar a los estudiantes. Los chatbots, cuando se usan de manera adecuada, permiten una interacción cercana, inmediata y accesible, reduciendo la carga cognitiva que generan otras plataformas más rígidas (Winkler & Söllner, 2018). Incorporar esta lógica de interacción en un entorno académico puede marcar la diferencia entre sentirse abrumado y sentirse acompañado en el aprendizaje.

En síntesis, el problema a resolver es doble: por un lado, la dificultad persistente para pasar de las ecuaciones a sus gráficas; y por otro, la falta de una herramienta web gratuita, intuitiva y de fácil acceso que permita graficar funciones en un entorno conversacional, con la posibilidad de guardar avances o simplemente consultar de manera rápida. Resolver este vacío no solo apoyará el desarrollo académico de estudiantes y docentes, sino que también fortalecerá la gestión del conocimiento en la institución, asegurando que lo aprendido no se pierda, sino que quede como base para futuras generaciones.

### **Justificación**

El aprendizaje de las matemáticas suele ser un reto porque lo que se escribe en símbolos no siempre resulta fácil de imaginar. La posibilidad de ver una función dibujada en una gráfica ayuda a que estudiantes y docentes comprendan mejor los conceptos y encuentren

sentido a lo que están resolviendo. Diversos estudios coinciden en que el uso de herramientas digitales interactivas no solo incrementa la comprensión conceptual, sino también la motivación de los estudiantes al momento de enfrentarse a funciones y ecuaciones (Li, Schoenfeld & Arcavi, 2019).

En el contexto universitario colombiano esta necesidad se hace más evidente. Investigaciones recientes muestran que muchos alumnos presentan dificultades al pasar de lo algebraico a lo gráfico, especialmente en funciones lineales, y que el uso de software como GeoGebra facilita la interpretación gráfica y mejora la confianza de los estudiantes (Mora Miranda, 2022). De manera similar, se ha evidenciado que resolver problemas matemáticos con apoyo de TIC fortalece la comunicación de ideas abstractas y disminuye la ansiedad frente al cálculo (Gallo Duarte, 2013).

Además, los estudios sobre integración tecnológica en educación superior en América Latina señalan que las barreras más comunes no son únicamente de acceso a computadores, sino de facilidad de uso y pertinencia pedagógica.

Cuando los recursos son complejos, de pago o poco accesibles, los estudiantes tienden a abandonarlos (Delgado & Sánchez, 2018). Esto resalta la importancia de contar con una herramienta web gratuita, sencilla y disponible desde cualquier navegador, sin necesidad de instalaciones.

EcuPlotWeb responde directamente a esa problemática. Por un lado, ofrece un modo de usuario registrado para que estudiantes y docentes guarden sus gráficas y lleven un seguimiento de sus avances; por otro, integra un modo invitado que permite consultas rápidas sin necesidad de registro. Esta dualidad busca atender tanto la necesidad de continuidad académica como la espontaneidad del aprendizaje. Trabajos desarrollados en universidades colombianas demuestran que cuando los estudiantes pueden guardar y retomar lo ya hecho, se favorece la reflexión sobre el propio proceso de aprendizaje y se fortalece la motivación (Toro Ríos, 2024; Daza-Correo, 2019).

Finalmente, no se trata solo de lo técnico. Un sistema conversacional como EcuPlotWeb también aporta en el plano emocional: un chat donde se pueda escribir una función y recibir su gráfica al instante reduce la frustración, promueve la exploración y fomenta la curiosidad. Estudios recientes sobre chatbots en educación destacan que este tipo de interacción inmediata y accesible favorece la cercanía con el contenido y la disposición para aprender (Winkler & Söllner, 2018).

En suma, la justificación de este proyecto descansa en tres pilares: accesibilidad (gratuito y disponible en la web), continuidad (con registro e historial de gráficas) y humanización del aprendizaje matemático (interacción sencilla, cercana y libre de barreras). Así, EcuPlotWeb no solo facilitará el estudio individual, sino que también aportará a la gestión del conocimiento dentro de la institución, asegurando que los aprendizajes de hoy sean la base para los avances del mañana.

## Objetivos

### Objetivo General

Desarrollar un software web educativo que permita graficar funciones matemáticas en un entorno conversacional accesible, con opción de registro para guardar avances y un modo invitado para consultas rápidas, con el fin de apoyar a estudiantes y docentes en la comprensión de conceptos matemáticos mediante representaciones gráficas claras y comprensibles.

### Objetivos específicos

1. Diseñar la arquitectura del software web EcuPlotWeb utilizando lenguajes como Python, HTML, CSS y JavaScript, con el fin de integrar un sistema de graficación interactivo y accesible desde cualquier navegador.
2. Implementar un módulo de gestión de usuarios que permita el registro, inicio de sesión e historial de gráficas, además de un modo invitado, garantizando así tanto la continuidad del aprendizaje como la accesibilidad para consultas rápidas.

3. Crear un sistema conversacional tipo chat que reciba funciones matemáticas y genere graficas dinámicas en tiempo real, incorporando bibliotecas especializadas como Matplotlib y Sympy para asegurar precisión y facilidad de uso.

## Metodología

El proyecto EcuPlotWeb se desarrollará utilizando la metodología ágil Extreme Programming (XP). XP es especialmente adecuada para equipos pequeños, como el nuestro de dos integrantes, y para proyectos en los que los requisitos pueden cambiar o refinarse sobre la marcha. Entre sus ventajas están entregas frecuentes, retroalimentación constante, pruebas y diseño simple.

A continuación, los principios/prácticas de XP que aplicaremos:

- Iteraciones cortas: ciclos de desarrollo de trabajo con entregas parciales para recibir retroalimentación rápida.
- Pruebas automatizadas: escribir pruebas unitarias antes de implementar funcionalidades nuevas, para asegurar calidad y detectar fallos temprano.
- Diseño simple y refactorización continua: mantener el código limpio, eliminar lo que sobra, mejorar lo construido cuando sea necesario.
- Comunicación constante: entre los dos integrantes, y en medida de lo posible con usuarios o docentes para validar que lo que se desarrolla tiene sentido.
- Entrega frecuente de valor: cada iteración entregara una parte funcional del sistema (por ejemplo, graficador básico, módulo de usuario, chat, etc.), no esperar al final para poner todo.

### **Fases del proyecto bajo XP**

Podemos dividir nuestro semestre en 14 semanas en estas fases, cada una con iteraciones, entregas parciales y pruebas y refactorización:

- Planificación inicial y exploración de requerimientos
- Diseño arquitectónico y de interfaces
- Desarrollo del núcleo funcional (graficador, motor de funciones)

- Desarrollo de funcionalidades adicionales (módulo de usuario, historial, modo invitado, chat)
- Pruebas finales, correcciones, validaciones
- Despliegue y refinamiento/ajustes finales

## Requerimientos

### Requerimientos Funcionales

ID	Requerimiento Funcional
RF-01	El sistema debe permitir a los invitados ingresar una función matemática y visualizar la gráfica con controles de zoom y pan.
RF-02	El sistema debe permitir a los visitantes registrarse con correo y contraseña, validando complejidad de la clave y correo único.
RF-03	El sistema debe permitir a los usuarios registrados iniciar sesión con credenciales válidas y mostrar error en caso contrario.
RF-04	El sistema debe enviar un correo con enlace de verificación para activar la cuenta de usuario
RF-05	El sistema debe permitir habilitar autenticación de dos factores (2FA) mediante app TOTP y códigos de recuperación.
RF-06	El sistema debe permitir a los <b>usuarios registrados</b> graficar funciones y guardarlas en su historial.
RF-07	El sistema debe permitir personalizar rangos (x,y) y estilos de la gráfica (color, grosor, grid).
RF-08	El sistema debe mostrar un historial de gráficas con búsqueda y filtrado por texto, fecha o etiquetas.

RF-09	El sistema debe permitir re-graficar y editar funciones del historial, generando nuevas versiones
RF-10	El sistema debe permitir eliminar funciones del historial con opción de recuperación durante 7 días
RF-11	El sistema debe permitir exportar gráficas en formatos PNG o SVG con metadatos de la función.
RF-12	El sistema debe soportar navegación por teclado y lectura por lectores de pantalla en los formularios principales
RF-13	El sistema debe renderizar funciones estándar en menos de 1.5 segundos en equipos promedio.
RF-14	El sistema debe permitir al administrador activar o desactivar cuentas de usuarios.
RF-15	El sistema debe permitir al administrador consultar un log de auditoría con acciones críticas, inicios fallidos y cambios de estado

### Requisitos no Funcionales

ID	Requerimiento No Funcional
RNF-01	Seguridad: Las contraseñas deben almacenarse en formato cifrado (hash seguro con sal).
RNF-02	Seguridad: El enlace de verificación debe expirar en un rango de 15–30 minutos y ser de un solo uso.
RNF-03	Seguridad: El login debe implementar bloqueo progresivo ante múltiples intentos fallidos para prevenir ataques de fuerza bruta.

RNF-04	Rendimiento: El renderizado de funciones debe ser fluido, optimizando memoria y evitando fugas.
RNF-05	Disponibilidad: El sistema debe estar accesible al menos el 99% del tiempo.
RNF-06	Usabilidad: La interfaz debe ser intuitiva, con mensajes claros de error y confirmación.
RNF-07	Compatibilidad: El sistema debe ser accesible desde navegadores modernos (Chrome, Firefox, Edge, Safari).
RNF-08	Accesibilidad: Cumplir parcialmente con WCAG 2.1 AA en formularios y navegación básica.
RNF-09	Escalabilidad: El sistema debe soportar al menos 500 usuarios concurrentes en la fase inicial.
RNF-10	Mantenibilidad: El código debe estar documentado y seguir buenas prácticas de arquitectura modular.
RNF-11	Privacidad: Los datos personales de los usuarios deben cumplir con normativas de protección de datos (ej. GDPR/ley local).
RNF-12	Portabilidad: La aplicación debe poder desplegarse tanto en entornos locales como en la nube.

## Casos de Uso

### Actores

- Usuario Invitado (UI): accede sin cuenta para graficar sin historial.
- Usuario Registrado (UR): accede con cuenta; tiene historial y opciones de cuenta.
- Sistema de Correo (SC): servicio externo para verificación y recuperación.

- Proveedor 2FA (2FA): servicio externo de códigos en un solo uso.
- Motor de Graficacion (MG): componente que valida, parsea y renderiza.
- Base de Datos (BD): persistencia de usuarios e historial.

## Catalogo de Casos de Uso

### **CU-01 Registrar cuenta.**

**Actor principal:** UR

**Descripción:** Crear una cuenta con correo y contraseña para habilitar historial y seguridad adicional.

**Precondiciones:** No haber iniciado sesión.

**Disparador:** El usuario pulsa “Crear cuenta”.

**Flujo principal:**

1. El sistema solicita correo y contraseña (y confirmación de contraseña).
2. Valida formato y unicidad del correo.
3. Cifra la contraseña y crea el registro en BD con estado “no verificado”.
4. Envía correo de verificación (SC) con enlace/código.

Muestra mensaje “Revisa tu correo para verificar tu cuenta”.

**Alternos/Excepciones.**

A1. Correo ya registrado → informar y sugerir “Recuperar contraseña”.

E1. Falla SC → reintento o “enviar de nuevo más tarde”.

Postcondiciones: Cuenta creada en estado “no verificado”.

### **CU-02 Verificar correo electrónico**

**Actor principal:** UR

**Descripción:** Activar la cuenta a través del enlace/código recibido por correo.

**Precondiciones:** Cuenta creada (CU-01).

**Disparador:** Usuario abre enlace de verificación.

**Flujo principal:**

1. El sistema recibe token/código de verificación.
2. Valida vigencia e integridad del token.
3. Cambia estado de usuario a “verificado”.
4. Inicia sesión y redirige al panel.

**Excepciones:** Token inválido/expirado → ofrecer reenviar verificación.

**Postcondiciones:** Usuario “verificado” y autenticado.

#### **CU-03 Iniciar sesión (con 2FA si está activo)**

**Actor principal:** UR

**Descripción:** Autenticación con correo y contraseña; si el usuario activó 2FA, se solicita código temporal.

**Precondiciones:** Usuario “verificado”.

**Disparador:** Usuario pulsa “Iniciar sesión”.

#### **Flujo principal:**

1. El sistema solicita correo y contraseña.
2. Valida credenciales.
3. Si 2FA **activado**: solicita código (app/correo/SMS según configuración).
4. Valida el código 2FA.
5. Inicia sesión y muestra panel.

#### **Alternos/Excepciones:**

A1. Contraseña incorrecta → informar, permitir reintentos limitados.

A2. 2FA desactivado → omitir paso 3.

E1. Código 2FA inválido/expirado → reintento o “Usar código de respaldo”.

**Postcondiciones:** Sesión iniciada; token/ cookie segura emitida.

#### **CU-04 Configurar 2FA (activar/desactivar)**

**Actor principal:** UR

**Descripción:** Activar 2FA (generar secreto/QR o elegir canal) o desactivarlo con verificación de contraseña.

**Precondiciones:** Sesión iniciada.

**Disparador:** Usuario abre “Seguridad de la cuenta”.

**Flujo principal:**

1. El sistema muestra opciones de 2FA (app TOTP / correo / SMS\*).
2. Para activar TOTP: muestra QR y solicita código de verificación.
3. Guarda estado 2FA=activo y genera **códigos de respaldo**.
4. Para desactivar: solicita contraseña y confirma.

**Excepciones:** Código TOTP inválido → reintentar.

**Notas:** \*SMS es opcional según costos/regulación.

#### **CU-05 Recuperar contraseña**

**Actor principal:** UR

**Descripción:** Restablecer contraseña mediante enlace temporal enviado al correo.

**Precondiciones:** Usuario registrado.

**Disparador:** “¿Olvidaste tu contraseña?”.

**Flujo principal:**

1. Ingresar correo.
2. Generar token de recuperación y enviar (SC).
3. Usuario abre enlace y define nueva contraseña.
4. Invalidar tokens previos e iniciar sesión.

**Excepciones:** Correo no registrado → mensaje genérico por seguridad.

#### **CU-06 Usar modo invitado**

**Actor principal:** UI

**Descripción:** Acceso inmediato a la herramienta de graficación sin registro.

**Precondiciones:** Ninguna.

**Disparador:** Usuario elige “Entrar como invitado”.

**Flujo principal:**

1. El sistema muestra el módulo de graficación.
2. Permite ingresar función y parámetros básicos.
3. Renderiza sin guardar historial.

**Postcondiciones:** Sesión temporal sin persistencia de datos personales.

**CU-07 Graficar función (invitado)**

**Actor principal:** UI (incluye MG)

**Descripción:** Ingresar una función y visualizarla con interacciones básicas.

**Precondiciones:** CU-06.

**Disparador:** Usuario pulsa “Graficar”.

**Flujo principal:**

1. Validar sintaxis (incluye **CU-14 Validar función**).
2. Generar datos (rango por defecto o configurado).
3. Renderizar gráfica interactiva (zoom, paneo, tooltip).
4. Permitir ajustar rango/precisión y re-renderizar.

**Excepciones:**

E1. Sintaxis inválida → mensaje claro y ejemplo de formato.

E2. Cálculo no converge/división por cero → resaltar dominio inválido.

**Postcondiciones:** Visualización activa; no se guarda historial.

**CU-08 Graficar función (registrado) incluye Guardar en historial**

**Actor principal:** UR (incluye MG y BD)

**Descripción:** Graficar y registrar la función en el historial del usuario.

**Precondiciones:** Sesión iniciada (CU-03).

**Disparador:** Usuario pulsa “Graficar”.

**Flujo principal:**

1. Validar sintaxis (CU-14).
2. Generar datos y renderizar.
3. Registrar entrada en **Historial** (función, fecha, parámetros).
4. Permitir renombrar/anotar la entrada.

**Excepciones:** BD no disponible → mostrar gráfica, pero encolar guardado/reintento.

#### **CU-09 Consultar y filtrar historial**

**Actor principal:** UR

**Descripción:** Buscar entradas por texto, fecha, tipo de función o etiquetas; ordenar y paginar.

**Precondiciones:** Historial existente.

**Disparador:** Usuario abre “Historial”.

#### **Flujo principal:**

1. Mostrar lista paginada con búsqueda y filtros.
2. Permitir ordenar (reciente, nombre, tipo).
3. Acciones: **Ver/Volver a graficar** (CU-10), **Renombrar**, **Eliminar**.

**Excepciones:** Sin resultados → “No hay coincidencias” y sugerencias de filtro.

#### **CU-10 Reusar una función del historial (volver a graficar)**

**Actor principal:** UR

**Descripción:** Cargar una entrada del historial, ajustar parámetros y re-graficar.

**Precondiciones:** CU-09.

**Disparador:** Usuario selecciona “Abrir en graficador”.

#### **Flujo principal:**

1. Cargar función y parámetros guardados.
2. Renderizar.
3. Permitir editar función/rangos y guardar como **nueva** entrada.

**Postcondiciones:** Nueva entrada opcional en historial.

### **CU-11 Exportar/Descargar gráfica**

**Actor principal:** UI/UR

**Descripción:** Exportar la visualización como imagen (PNG/SVG) o datos (CSV) bajo demanda.

**Precondiciones:** Existe una gráfica renderizada.

**Disparador:** Usuario pulsa “Exportar”.

**Flujo principal:**

1. Elegir formato (PNG/SVG/CSV).
2. Generar archivo y descargar.

**Excepciones:** Exportar bloqueado por pop-up → instrucción para permitir descarga.

### **CU-12 Gestionar perfil**

**Actor principal:** UR

**Descripción:** Cambiar nombre visible, correo\* y contraseña; gestionar privacidad.

**Precondiciones:** Sesión iniciada.

**Disparador:** Usuario abre “Perfil”.

**Flujo principal:**

1. Mostrar datos editables.
2. Guardar cambios; si cambia **correo**, requerir **re-verificación** (CU-02).

**Excepciones:** Contraseña actual incorrecta → informar y reintento.

### **CU-13 Cerrar sesión**

**Actor principal:** UR

**Descripción:** Finalizar sesión y revocar tokens activos del navegador.

**Precondiciones:** Sesión iniciada.

**Disparador:** Usuario pulsa “Cerrar sesión”.

**Flujo principal:**

1. Invalidar sesión en servidor.

2. Borrar credenciales locales.
3. Redirigir a inicio.

**Postcondiciones:** Sesión finalizada con seguridad.

#### **CU-14 Validar función (incluido por CU-07/08/10)**

**Actor principal:** MG

**Descripción:** Comprobar sintaxis, dominios y parámetros antes de graficar.

**Precondiciones:** Cadena de función disponible.

**Disparador:** Llamado desde graficación.

**Flujo principal:**

1. Parsear expresión y detectar variables permitidas.
2. Validar dominios (p. ej., raíces de pares con negativos, división por cero).
3. Normalizar parámetros (rango, paso).

**Excepciones:** Expresión no soportada → devolver mensaje y ejemplos.

#### **Pruebas Unitarias**

En el proyecto EcuPlot Web se utiliza el framework pytest como herramienta principal para la ejecución de pruebas automatizadas. Pytest permite definir casos de prueba simples y legibles en Python, agruparlos por módulos (por ejemplo: servicios de contraseñas, endpoints de graficación, rutas de roles, validación de email y 2FA, exportación CSV/JSON, sistema de tickets, historial de gráficas, etc.) y ejecutarlos de forma masiva desde la línea de comandos, obteniendo un resumen claro de cuántas pruebas pasan, fallan o se omiten.

```
Ejemplo: Verify logging is configured when app is created.
- Tickets Crud: 3 pruebas (P:3 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
  Ejemplo: Encadena llamadas (POST /api/account/requests, GET /api/account/requests?page=1&page_size=5) para validar comportamientos de tickets crud.
- Twofa Routes: 2 pruebas (P:2 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
  Ejemplo: Usuario sin 2FA debe retornar enabled=False.
- Validate Service: 18 pruebas (P:18 F:0 S:0) | Alcance: Prueba unitaria | Metodologías: Caja blanca (lógica aislada)
  Ejemplo: Email debe convertirse a minúsculas.
- Verify Email: 4 pruebas (P:4 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
  Ejemplo: Valida el escenario 'verify email flow' dentro de verify_email.
```

```

Ejemplo: Verify logging is configured when app is created.
- Tickets Crud: 3 pruebas (P:3 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
Ejemplo: Encadena llamadas (POST /api/account/requests, GET /api/account/requests?page=1&page_size=5) para validar comportamientos de tickets crud.
- Twofa Routes: 2 pruebas (P:2 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
Ejemplo: Usuario sin 2FA debe retornar enabled=False.
- Validate Service: 18 pruebas (P:18 F:0 S:0) | Alcance: Prueba unitaria | Metodologías: Caja blanca (lógica aislada)
Ejemplo: Email debe convertirse a minúsculas.
- Verify Email: 4 pruebas (P:4 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
Ejemplo: Valida el escenario 'verify email flow' dentro de verify_email.

```

```

- Passwords Service: 30 pruebas (P:30 F:0 S:0) | Alcance: Prueba de integración, Prueba unitaria | Metodologías: Caja blanca (capas internas controladas), Caja blanca (lógica aislada)
Ejemplo: None debe retornar_error de política.
- Plot Endpoints: 5 pruebas (P:5 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
Ejemplo: Verifica que la petición POST /api/plot responda correctamente dentro de la funcionalidad de plot endpoints.
- Plot History Autotags: 13 pruebas (P:13 F:0 S:0) | Alcance: Prueba unitaria | Metodologías: Caja blanca (lógica aislada)
Ejemplo: Valida el escenario 'classify expression detects primary categories' dentro de plot history autotags.
- Plot Tags Coverage: 19 pruebas (P:19 F:0 S:0) | Alcance: Prueba de integración, Prueba unitaria | Metodologías: Caja blanca (capas internas controladas), Caja blanca (lógica aislada)
Ejemplo: Debe manejar None.
- Rate Limiting: 14 pruebas (P:14 F:0 S:0) | Alcance: Prueba unitaria | Metodologías: Caja blanca (lógica aislada)
Ejemplo: Prueba que requests dentro del límite funcionan correctamente.
- Role Requests: 1 pruebas (P:1 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
Ejemplo: Encadena llamadas (POST /api/role-requests, GET /api/role-requests/me, POST /api/role-requests) para validar comportamientos de role requests.
- Roles Learning Errors: 6 pruebas (P:6 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
Ejemplo: Debe enviar notificación al crear solicitud de rol.
- Roles Routes: 2 pruebas (P:2 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
Ejemplo: Sin solicitud debe retornar request=None.
- SSE Routes: 2 pruebas (P:2 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
Ejemplo: Sin token debe retornar 401.
- Structured Logging: 17 pruebas (P:17 F:0 S:0) | Alcance: Prueba de integración, Prueba funcional / integración, Prueba unitaria | Metodologías: Caja blanca (capas internas controladas), Caja blanca (lógica aislada), Caja negra (interacción HTTP)
Ejemplo: Verify logging is configured when app is created.
- Tickets Crud: 3 pruebas (P:3 F:0 S:0) | Alcance: Prueba funcional / integración | Metodologías: Caja negra (interacción HTTP)
Ejemplo: Encadena llamadas (POST /api/account/requests, GET /api/account/requests?page=1&page_size=5) para validar comportamientos de tickets crud.

```

En las pruebas ejecutadas se observan distintos tipos de pruebas y metodologías

**Pruebas unitarias sobre lógica aislada (**caja blanca: capas internas controladas y lógica de validación**).**

Pruebas funcionales e integración sobre la API HTTP (caja negra: interacción real con endpoints como /api/plot, /api/role-requests, /api/account/requests, exportaciones, rutas de frontend y health checks).

Casos específicos para seguridad y autenticación (Passwords\_Service, Twofa Routes, Validate\_Service, Verify Email), control de tráfico (Rate Limiting) y consistencia de datos (History Mutations, Plot History Autoguess, Event Stream, Structured Logging).

## Análisis de resultados

Los resúmenes mostrados para cada módulo indican patrones del tipo:

X pruebas (P:X F:0 S:0) es decir:

- P: número de pruebas pasadas.

- F: número de pruebas fallidas (0).
- S: número de pruebas saltadas (0).

El hecho de que en todos los servicios listados el valor de F sea 0 y S sea 0 significa que **todas las pruebas definidas se ejecutaron y pasaron correctamente**. Esto sugiere:

La lógica crítica de negocio (generación de gráficas, manejo de historiales, solicitudes de roles, tiques, etc.) se comporta según lo esperado en los escenarios probados.

Los mecanismos de seguridad (gestión de contraseñas, verificación de email, validación de 2FA) funcionan sin errores en las condiciones definidas en los casos de prueba.

La API HTTP y las rutas del frontend responden correctamente a las peticiones encadenadas que se usan en el flujo real de la aplicación (por ejemplo, creación de solicitudes, paginación de historiales, exportación de datos).

El logging estructurado y los servicios de monitoreo básicos se crean y configuran adecuadamente al iniciar la aplicación, lo que facilita depuración y trazabilidad.

### Historias de Usuario

ID	Rol	Historia de Usuario
HU-01	Invitado	Ingresar una función matemática y visualizar la gráfica de forma interactiva (zoom/pan).
HU-02	Visitante	Registrarse con correo y contraseña para guardar historial de gráficas.
HU-03	Usuario Registrado	Iniciar sesión con credenciales válidas para acceder al historial de gráficas.
HU-04	Usuario	Verificar su correo mediante enlace para activar la cuenta.
HU-05	Usuario	Habilitar 2FA con app TOTP para mayor seguridad en el inicio de sesión.
HU-06	Usuario Registrado	Ingresar una función y graficarla con opción de guardarla en historial.
HU-07	Usuario	Ajustar rangos (x,y) y estilo de la gráfica para personalizar la visualización.
HU-08	Usuario Registrado	Ver y buscar el historial de gráficas para reutilizar funciones previas.
HU-09	Usuario Registrado	Re-graficar o editar entradas del historial creando nuevas versiones.
HU-10	Usuario Registrado	Eliminar entradas del historial con opción de recuperación en 7 días.
HU-11	Usuario	Exportar la gráfica en PNG o SVG para usarla en informes.

HU-12	Usuario	Operar el flujo principal con lector de pantalla/teclado (accesibilidad básica).
HU-13	Usuario	Obtener gráficas que carguen en <1.5s para funciones estándar y sin pérdidas de memoria.
HU-14	Administrador	Activar o desactivar cuentas de usuario para gestionar accesos.
HU-15	Administrador	Ver un log básico de auditoría con acciones críticas, inicios fallidos y cambios de estado.

### Product Backlog

ID	Épica/Feature	HU Relacionadas	Descripción breve	Prioridad	SP	Dependencias	Criterio de Hecho (DoD)
PB-01	A&V – Registro	HU-02	Registro con validación y email	Must	3	Email	PR + pruebas, manejo de errores, logs base
PB-02	A&V – Login	HU-03	Inicio de sesión seguro	Must	3	HU-02/04	PR + unit tests + bloqueo por intentos
PB-03	A&V – Verificación correo	HU-04	Activación por enlace	Must	2	Email	Token expira, reenvío funcional
PB-04	A&V – 2FA TOTP	HU-05	Doble factor opcional	Should	5	HU-03	QR, códigos recuperación, pruebas

PB-05	GG – Graficar Invitado	HU-01	Render básico sin guardar	Must	5	Motor gráfico	95% casos felices + errores comunes
PB-06	GG – Graficar Usuario	HU-06	Render + “Guardar”	Must	5	PB-05	Persistencia mínima validada
PB-07	GG – Parámetros y Estilo	HU-07	Rango, color, grid	Should	3	PB-06	UI reactiva + validaciones
PB-08	HIS – Listar/Buscar	HU-08	Historial con filtros	Must	5	PB-06	Índices BD + paginación
PB-09	HIS – Re-graficar/Editar	HU-09	Reusar y versionar	Should	3	PB-08	Versionado no destructivo
PB-10	HIS – Eliminar	HU-10	Soft delete + recuperación	Should	2	PB-08	Confirmación + papelera 7 días
PB-11	GG – Exportar imagen	HU-11	PNG/SVG con metadatos	Should	3	PB-06	Control de resolución + pruebas
PB-12	UX – Accesibilidad	HU-12	WCAG 2.1 AA básico	Must	3	UI base	Auditoría ARIA + tab order
PB-13	UX – Rendimiento	HU-13	TTI < 1.5s estándar	Must	5	PB-05/06	Métricas en entorno de prueba
PB-14	ADM – Activar/Desactivar	HU-14	Gestión de estado	Must	3	A&V	Cambios auditados y reversibles
PB-15	OBS – Auditoría mínima	HU-15	Log de eventos clave	Should	3	PB-14	Filtros por fecha/acción, export CSV

## Funcionalidades

Funcionalidad	Descripción
---------------	-------------

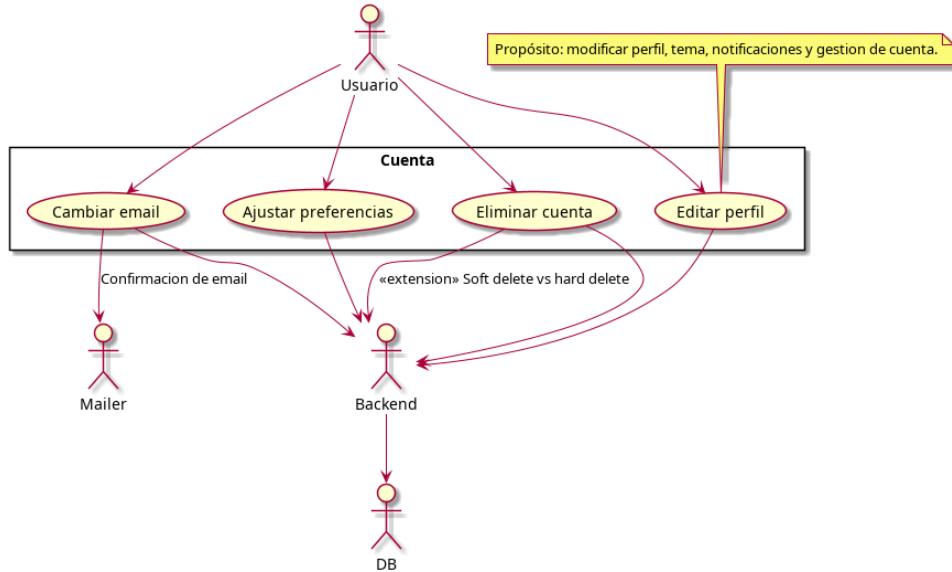
Graficar como invitado	Permite ingresar una función matemática y visualizar su gráfica de manera interactiva sin registrarse.
Registro de usuarios	Permite a los visitantes crear una cuenta con correo y contraseña para guardar su historial de gráficas.
Inicio de sesión	Permite a los usuarios registrados acceder a su cuenta con credenciales válidas.
Verificación de correo electrónico	Envía un enlace de verificación al correo del usuario para activar la cuenta.
Autenticación en dos pasos (2FA)	Ofrece mayor seguridad en el inicio de sesión mediante códigos TOTP y de recuperación.
Graficar como usuario registrado	Permite a los usuarios registrados ingresar funciones, graficarlas y guardarlas en el historial.
Personalización de rangos y estilo	Permite modificar los rangos de los ejes y el estilo visual de la gráfica (colores, grosor, grid).
Historial de gráficas	Muestra al usuario registrado una lista paginada de funciones graficadas previamente con búsqueda.
Re-graficar y editar historial	Permite reutilizar funciones del historial, re-graficarlas o editarlas creando nuevas versiones.
Eliminar historial	Permite eliminar funciones del historial con opción de recuperación en 7 días.
Exportar gráfica	Permite descargar la gráfica en formato PNG o SVG con metadatos de la función.
Accesibilidad básica	Asegura que la aplicación pueda ser usada con teclado y lectores de pantalla (WCAG 2.1 AA).

Rendimiento optimizado	Garantiza que las gráficas estándar se rendericen en menos de 1.5s y sin fugas de memoria.
Gestión de usuarios (admin)	Permite al administrador activar o desactivar cuentas de usuarios.
Auditoría mínima	Ofrece un registro básico de acciones críticas, inicios fallidos y cambios en el sistema.

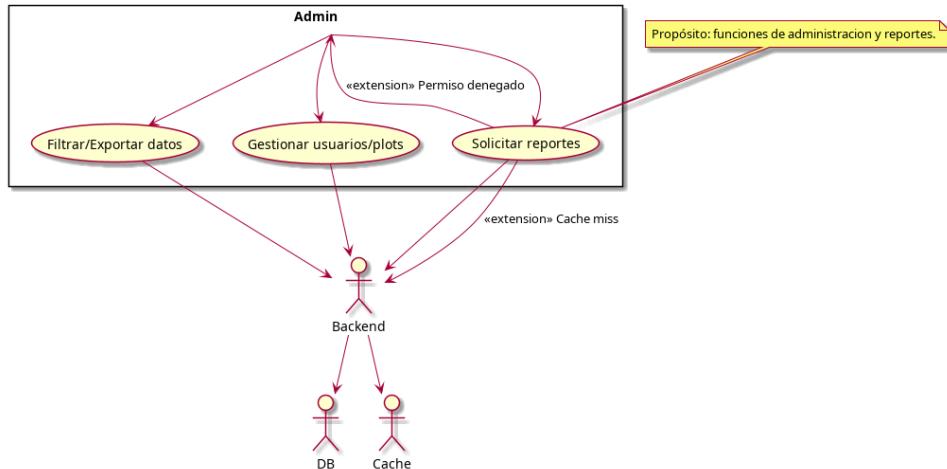
## Diagramas

### Casos de Uso

Caso de uso: Gestión de cuenta y preferencias

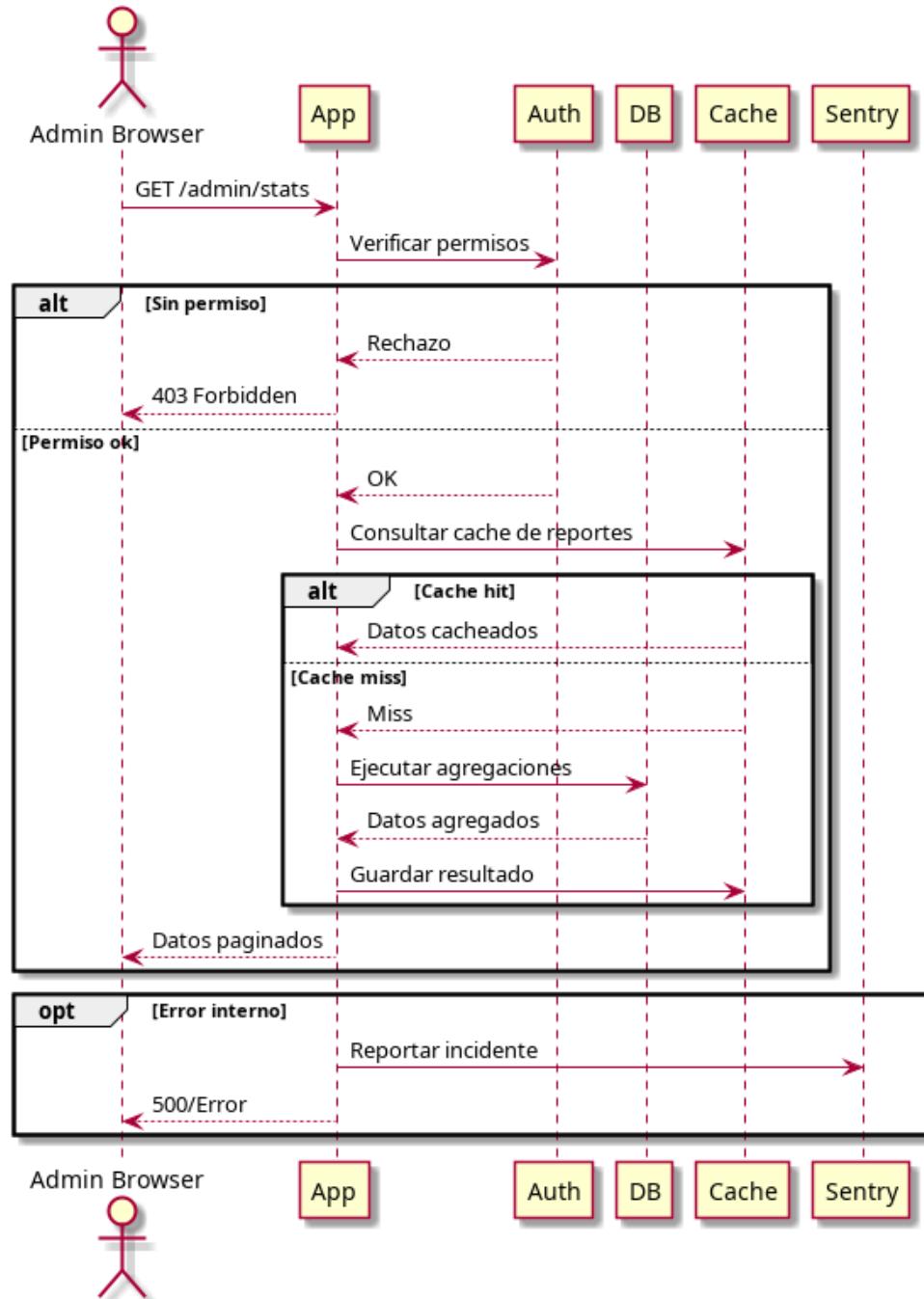


Caso de uso: Panel de administración — estadísticas y gestión

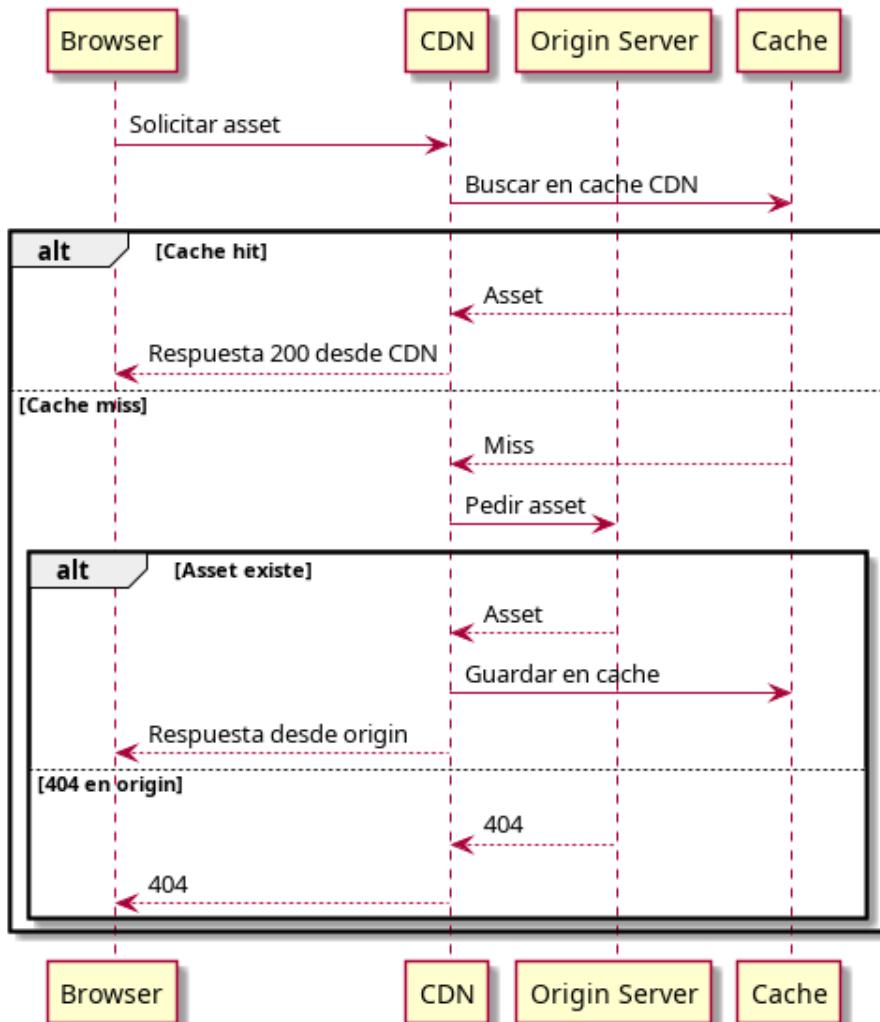


## Secuencia

Secuencia: Solicitud de estadísticas de admin (admin\_stats\_plots)

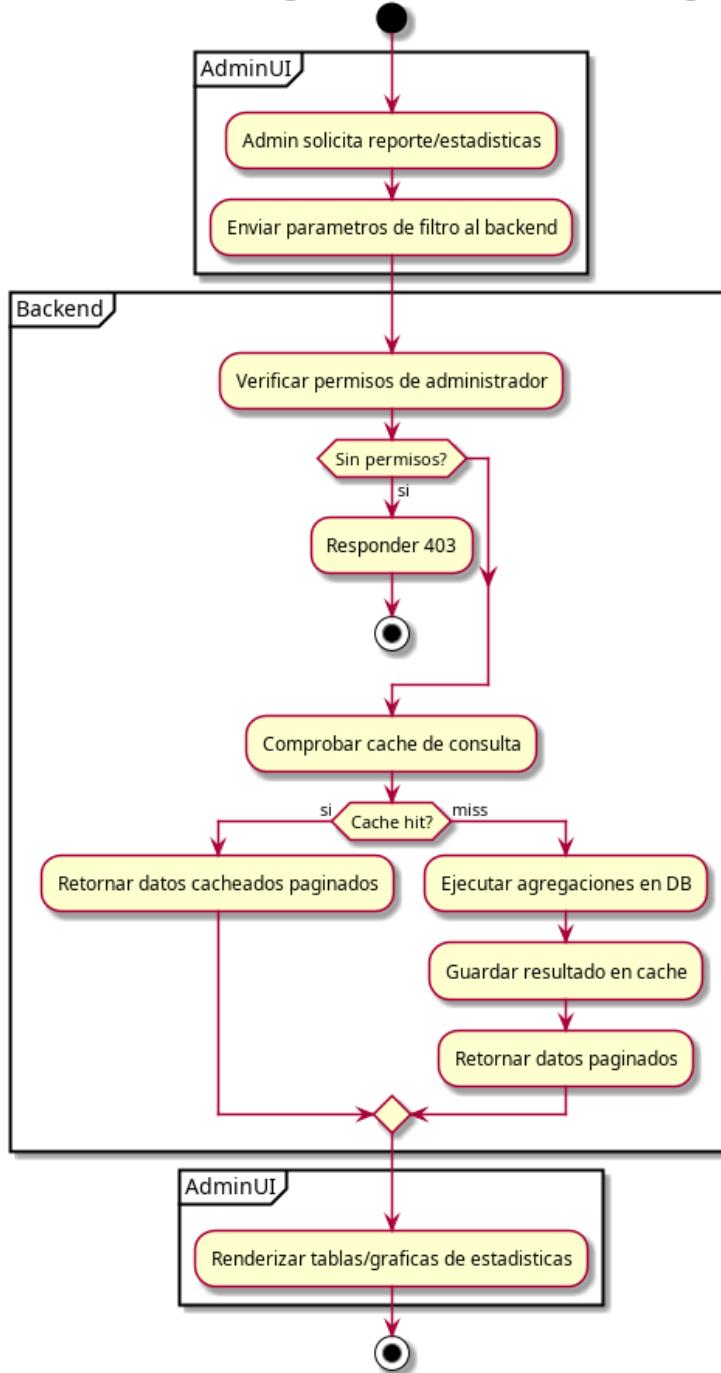


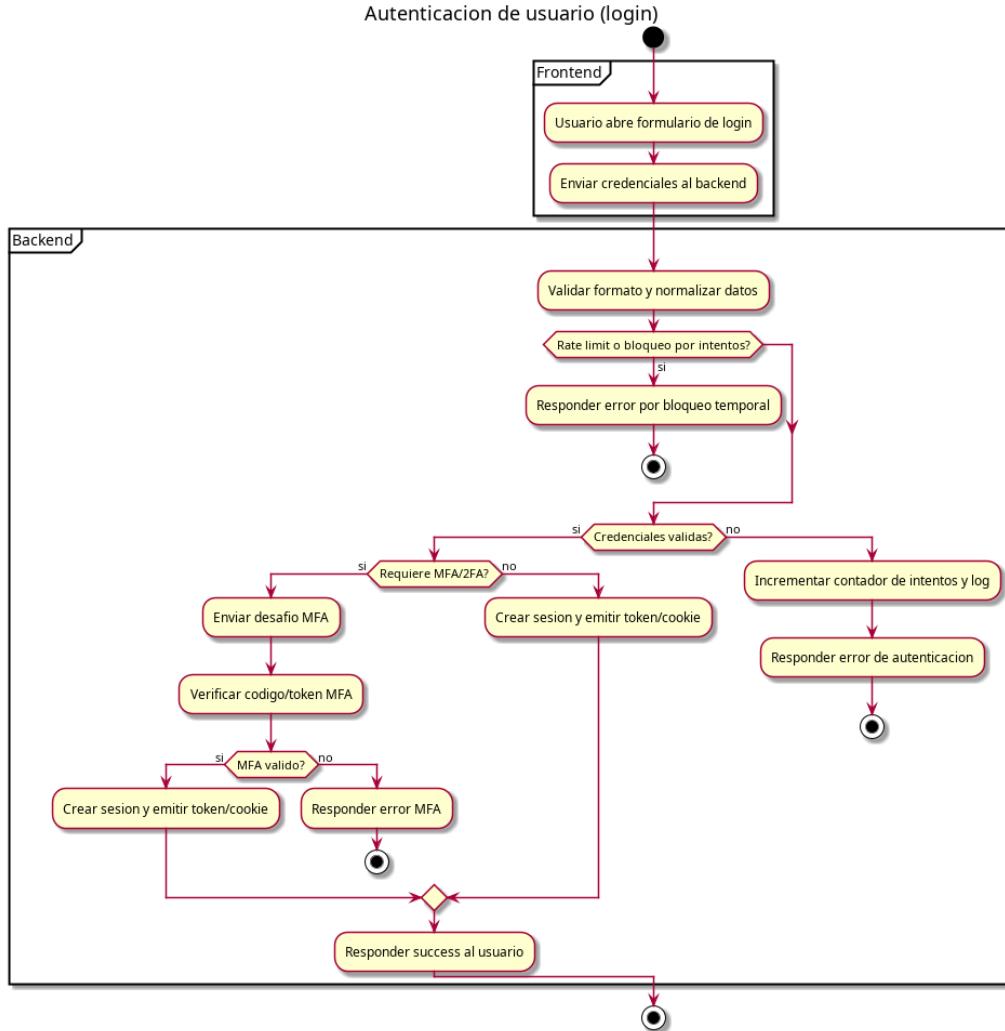
### Secuencia: Petición de assets estaticos / CDN



## Actividades

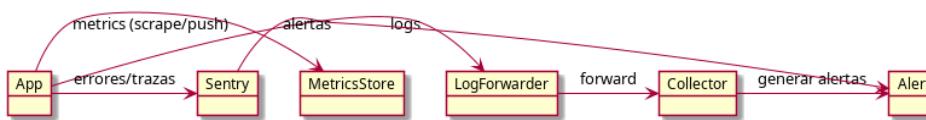
Panel de administracion: generacion de estadisticas de graficas



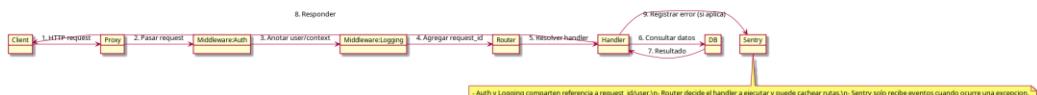


## Colaboración

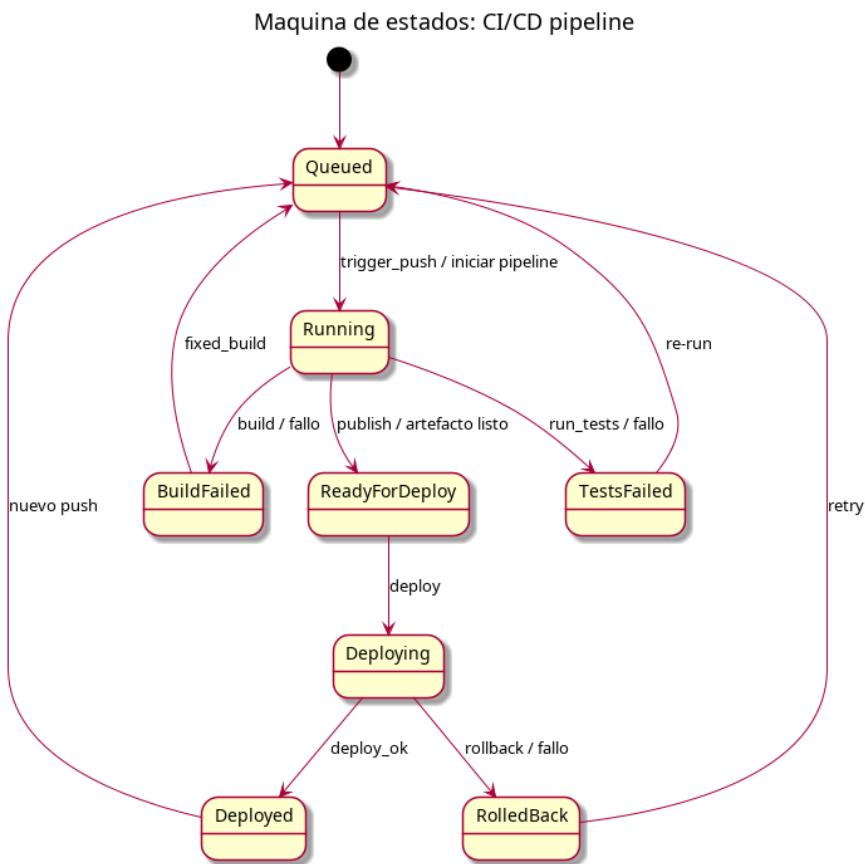
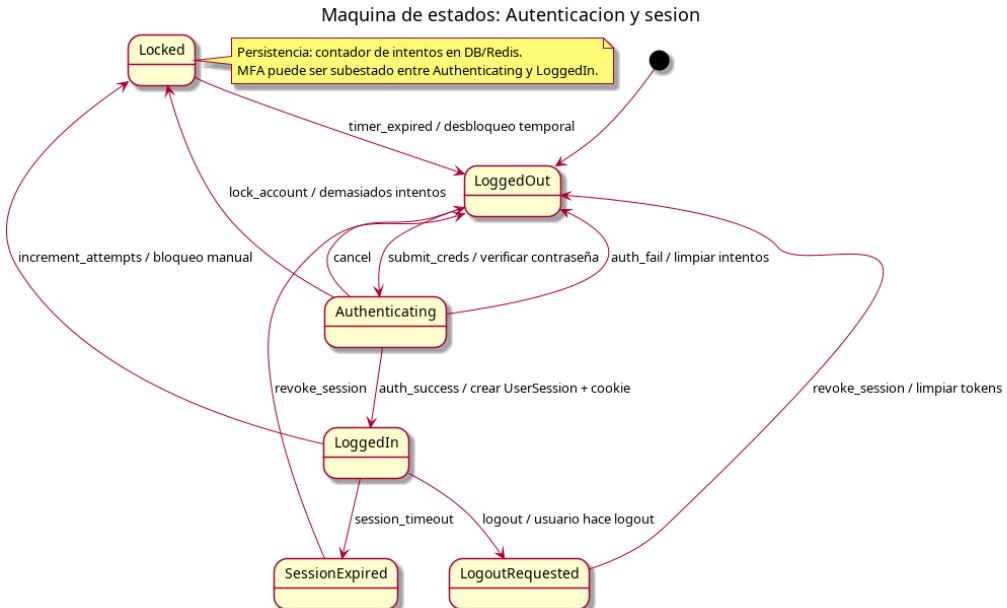
Colaboracion: Observabilidad y logging



Colaboracion: Request lifecycle / middleware

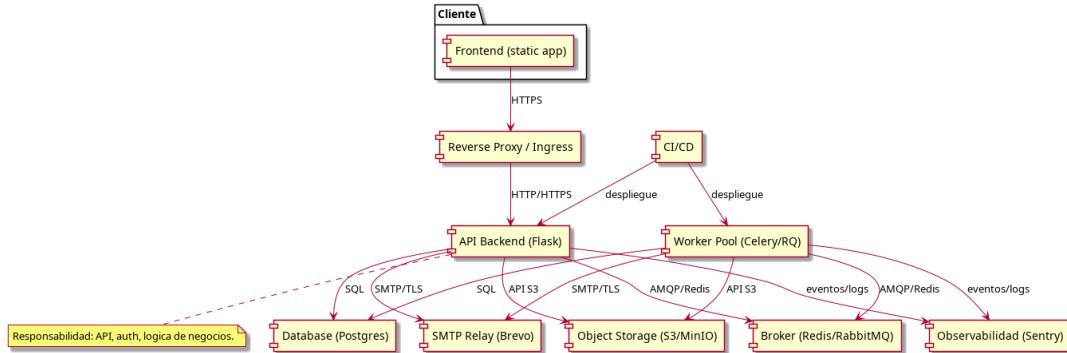


## Máquina de Estado

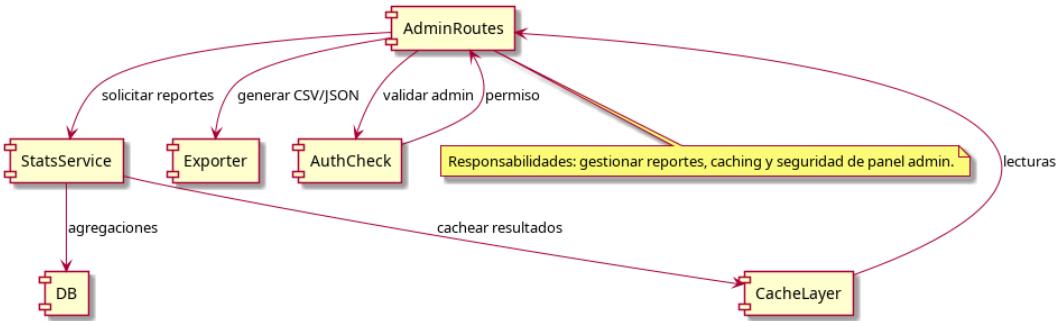


## Componentes

Componentes: Vision general del sistema

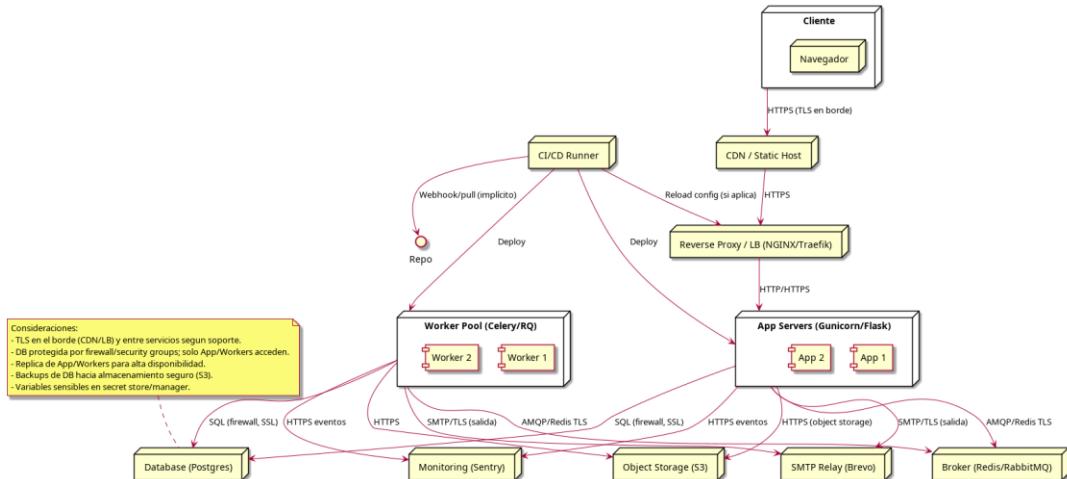


Componentes: Admin / Reporting

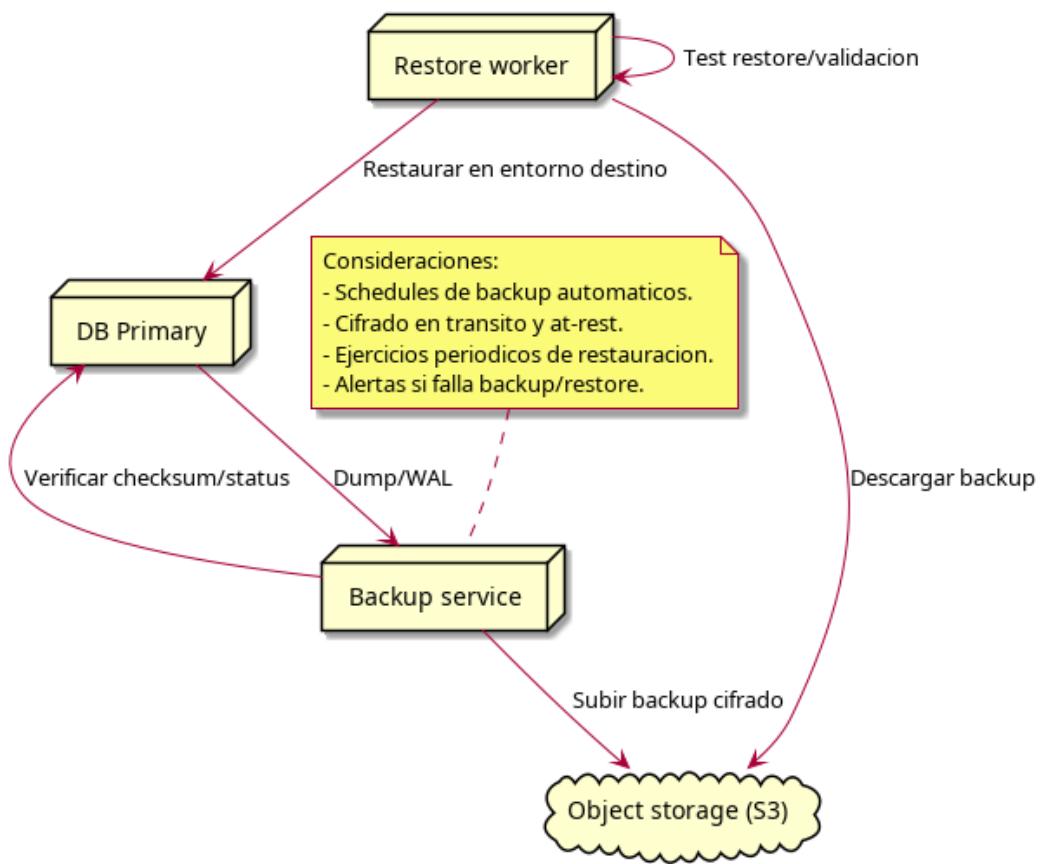


## Despliegue

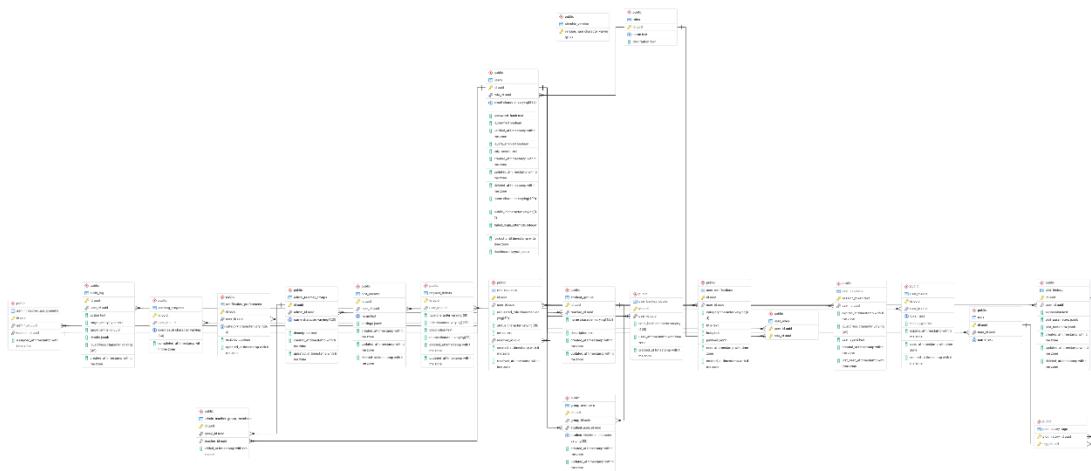
Despliegue: Arquitectura general (overview)

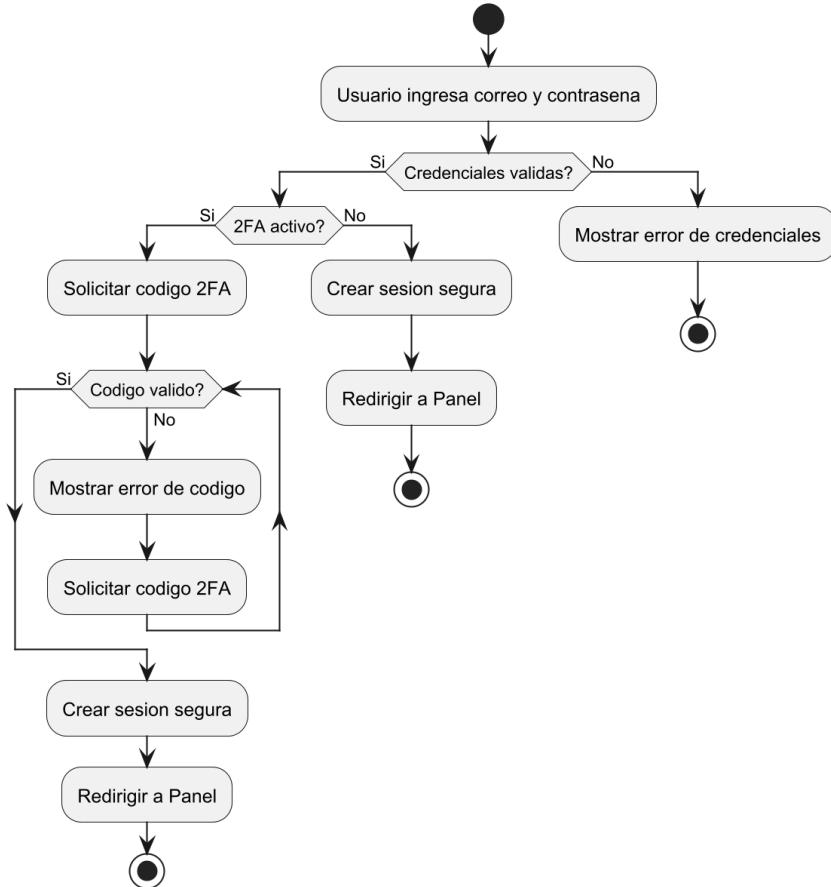
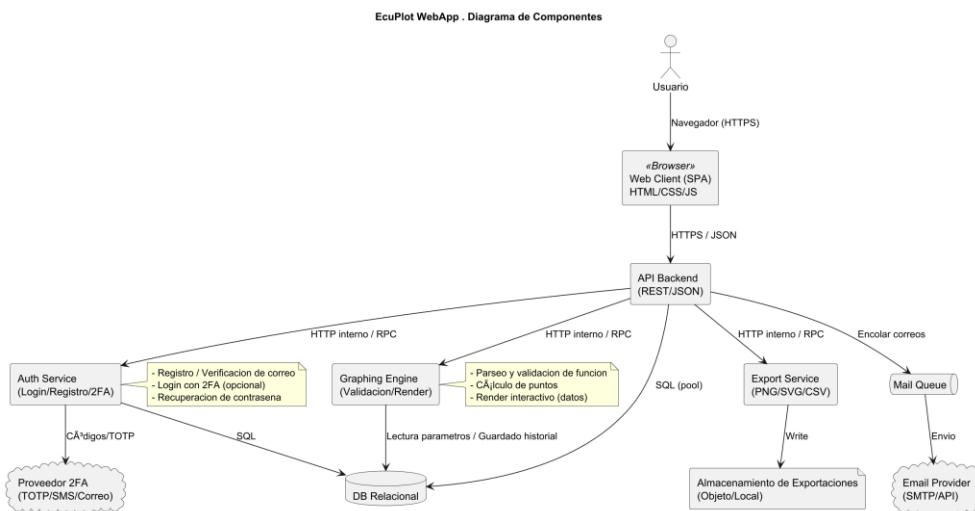


## Despliegue: Backup y recuperacion

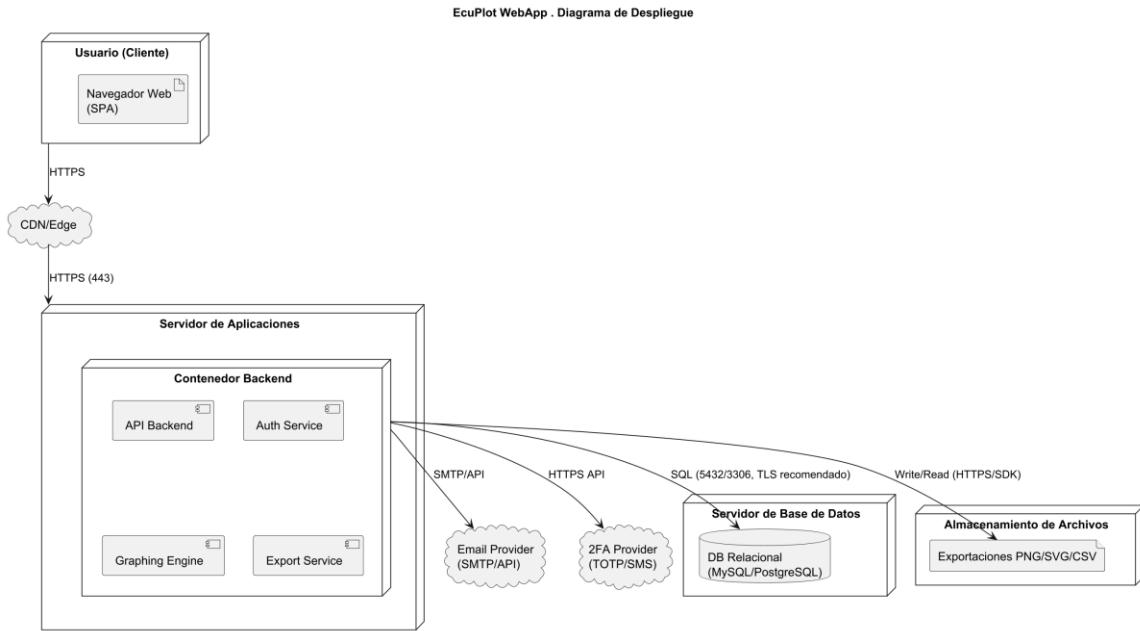


## Diagrama Entidad Relación

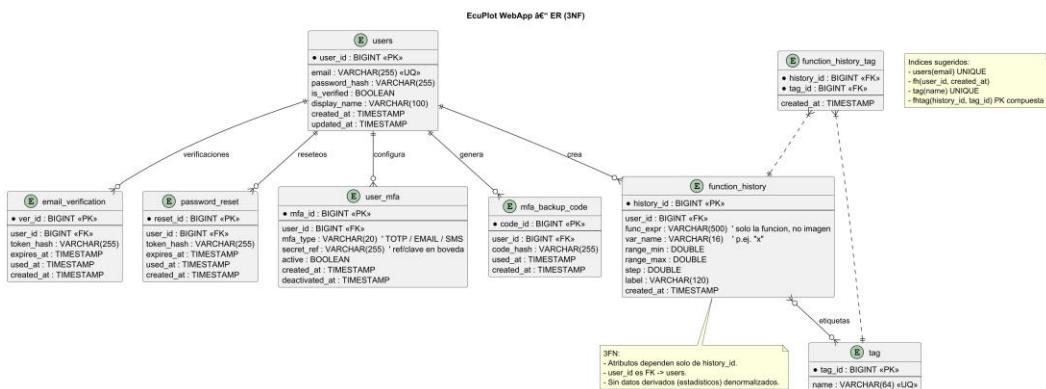


**Figura 1****EcuPlot - Actividad: Inicio de sesion con 2FA****Figura 2**

**Figura 3**



**Figura 4**



## Metodología XP aplicada al proyecto

### Principios adoptados

- Simplicidad: Implementar solo lo necesario para pasar las pruebas (MVP primero).
- Comunicación: Parejas de programación Mateo–Julian; daily breve asíncrona (tablero).
- Retroalimentación: Demostraciones quincenales al docente; issues con comentarios.
- Coraje: Refactorizaciones pequeñas y frecuentes; eliminación de deuda técnica temprana.
- Respeto: Revisión de código empática y tiempos realistas.

### Roles del Equipo

- Mateo Ibarra: Backend, librerías de graficacion, pruebas unitarias.
- Julian Cardenas: Frontend, base de datos, documentación y coordinación con el docente.

### Prácticas XP

- Programación en parejas: Sesiones de 90 min; rotación según modulo.
- Integración continua: Pipeline con tests + lint + build en cada commit a develop.
- Pruebas unitarias y de aceptación: Cobertura mínima 70% en módulos núcleo; pruebas Gherkin para HU críticas.
- Refactorización continua: Regla “Boy Scout”: dejar el código un poco mejor.
- Retroalimentación frecuente del cliente (docente): Demo y acta quincenal.

### Plan de Iteraciones (2 semanas c/u)

- Iteracion1 (Sem 1-2): requisitos y primeros diagramas.
- Iteracion2 (Sem 3-4): diseño BS +PoC graficacion.
- Iteracion3 (Sem 5-6): autenticación (registro/login/verificación)
- Iteracion4 (Sem 7-8): graficacion web + parámetros.

- Iteracion5 (Sem 9-10): historial completo.
- Iteracion6 (Sem 11-12): integración y despliegue completo.

### **Actas de Reunion**

#### **Acta de Reunión XP No. 1**

**Fecha:** 14/08/202

**Proyecto:** Ecoplot Web

**Tipo de reunión:** *Planning Game*, Reunión de planificación del proyecto.

**Participantes:** Julián Alejandro Cárdenas Santiago (Líder técnico), Carlos Mateo Cruz Ibarra (Desarrollador)

#### **Temas tratados:**

- Definición del MVP.
- Identificación de actores principales.
- Redacción de los casos de uso iniciales.

#### **Decisiones tomadas:**

- Validar 6 casos clave y épicas para la primera iteración.

#### **Compromisos:**

<b>Responsable</b>	<b>Tarea</b>	<b>Fecha límite</b>
Julián	Documentar casos de uso y priorizar épicas.	15/08/2025
Mateo	Crear diagrama de actores y sistema.	16/08/2025

#### **Observaciones:**

El alcance se centrará en la función de graficacion matemática básica y el almacenamiento de historial.

Inicio del proyecto *Ecoplot Web*. Definición de alcance, actores y casos de uso clave qué se va a hacer y por qué es importante.

### **Acta de Reunión XP No. 2**

**Fecha:** 21/08/2025

**Proyecto:** Ecoplot Web

**Tipo de reunión:** *Technical Review*, revisión técnica de arquitectura/datos.

**Participantes:** Julián Alejandro Cárdenas Santiago, Carlos Mateo Cruz Ibarra

#### **Temas tratados:**

- Diseño del modelo de datos preliminar.
- Relaciones entre usuarios, historial y funciones graficadas.

#### **Decisiones tomadas:**

- Implementar índices en historial para optimizar búsqueda.
- Aplicar normalización básica de tablas.

#### **Compromisos:**

<b>Responsable</b>	<b>Tarea</b>	<b>Fecha límite</b>
Mateo	Implementar modelo en base de datos PostgreSQL.	22/08/2025
Julián	Revisar consistencia y normalización.	23/08/2025

Discusión del modelo de datos preliminar, definir índices y aplicar normalización básica.

Esto dio una base coherente para todas las futuras funcionalidades. Como se estructura el sistema por dentro

### **Acta de Reunión XP No. 3**

**Fecha:** 28/08/2025

**Proyecto:** Ecoplot Web

**Tipo de reunión:** *Design Meeting*, Reunión de diseño de interfaz y experiencia.

**Participantes:** Julián Alejandro Cárdenas Santiago, Carlos Mateo Cruz Ibarra

**Temas tratados:**

- Prototipo de interfaz UI para login y registro.
- Flujo de autenticación con validación por correo.

**Decisiones tomadas:**

- Aceptar flujo de verificación por correo electrónico como requisito obligatorio.

**Compromisos:**

Responsable	Tarea	Fecha límite
Mateo	Diseñar vista responsive de login.	29/08/2025
Julián	Implementar endpoint de verificación.	30/08/2025

Definición del prototipo de login y registro, incluyendo verificación por correo, asegurando una experiencia coherente y segura. cómo interactuará el usuario con el sistema.

**Acta de Reunión XP No. 4**

**Fecha:** 04/09/2025

**Proyecto:** Ecoplot Web

**Tipo de reunión:** *Proof of Concept (PoC)*, Prueba de concepto técnica.

**Participantes:** Julián Alejandro Cárdenas Santiago, Carlos Mateo Cruz Ibarra

**Temas tratados:**

- Prueba de librerías de graficacion: Plotly y Matplotlib.
- Evaluación de rendimiento en entorno web.

**Decisiones tomadas:**

- Adoptar **Plotly.js** como motor principal.
- Incluir *fallback* estático con Matplotlib para exportación.

**Compromisos:**

Responsable	Tarea	Fecha límite
Julián	Integrar Plotly con backend Flask.	06/09/2025
Mateo	Probar exportación de gráficos en formato PNG.	07/09/2025

Se evaluó Plotly y Matplotlib para graficar funciones. Al elegir Plotly para web (y Matplotlib como respaldo) enfoque técnico **eficiente y viable**. Clave para reducir riesgos técnicos tempranos.

**Acta de Reunión XP No. 5**

**Fecha:** 11/09/2025

**Proyecto:** Ecoplot Web

**Tipo de reunión:** Daily XP + Debugging Session Reunión diaria / sesión de depuración.

**Participantes:** Julián Alejandro Cárdenas Santiago, Carlos Mateo Cruz Ibarra

**Temas tratados:**

- Errores de sintaxis en graficador.
- Retroalimentación de validación de expresiones matemáticas.

**Decisiones tomadas:**

- Implementar resaltada visual de posición del error en el campo de texto.

**Compromisos:**

Responsable	Tarea	Fecha límite
Mateo	Desarrollar módulo de resaltado en tiempo real.	12/09/2025
Julián	Revisar parser de expresiones.	13/09/2025

### Acta de Reunión XP No. 6

**Fecha:** 18/09/2025

**Proyecto:** Ecoplot Web

**Tipo de reunión:** *Iteration Review*

**Participantes:** Julián Alejandro Cárdenas Santiago, Carlos Mateo Cruz Ibarra

#### **Temas tratados:**

- Implementación del historial de búsquedas.
- Gestión de etiquetas y eliminación lógica (*soft delete*).

#### **Decisiones tomadas:**

- Definir paginación en el historial y aplicar *soft delete* para mantener trazabilidad.

#### **Compromisos:**

Responsable	Tarea	Fecha límite
Julián	Diseñar endpoint de historial con filtros.	19/09/2025

Responsable	Tarea	Fecha límite
Mateo	Crear interfaz con paginación dinámica.	20/09/2025

historial de búsquedas, etiquetas y eliminación lógica. validar que los módulos estaban cumpliendo el propósito del MVP. cierre técnico de iteración y verificación funcional.

### Acta de Reunión XP No. 7

**Fecha:** 25/09/2025

**Proyecto:** Ecoplot Web

**Tipo de reunión:** Retrospectiva XP

**Participantes:** Julián Alejandro Cárdenas Santiago, Carlos Mateo Cruz Ibarra

#### **Temas tratados:**

- Configuración de parámetros personalizados para gráficos.
- Estilos y preferencias del usuario.

#### **Decisiones tomadas:**

- Permitir guardar *presets* por usuario para reutilización de estilos.

#### **Compromisos:**

Responsable	Tarea	Fecha límite
Mateo	Implementar módulo de presets en frontend.	27/09/2025
Julián	Validar guardado en base de datos y endpoints.	28/09/2025

Evaluaron la personalización de parámetros de gráficos y la posibilidad de guardar presets.

### Revisión actas de iteración xp

<b>Tipo de reunión</b>	<b>Momento del ciclo</b>	<b>Propósito</b>
Planning Game	Inicio del proyecto	Definir alcance y prioridades.
Technical Review	Etapa temprana	Asegurar base técnica sólida.
Design Meeting	Etapa de diseño	Definir UI/UX funcional.
Proof of Concept	Validación técnica	Comprobar tecnologías.
Daily XP + Debugging	Iteración continua	Resolver bloqueos y errores.
Iteration Review	Fin de iteración	Revisar avances.
Retrospectiva XP	Cierre de ciclo	Mejorar prácticas y desempeño.

### Historial de Reuniones Semanales

<b>Semana</b>	<b>Fecha</b>	<b>Temas tratados</b>	<b>Decisiones/Acciones</b>
1	14/08/2025	Alcance MVP, actores, casos de uso	Validar 6 casos clave y épicas
2	21/08/2025	Modelo de datos preliminar	Índices en historial; normalización básica
3	28/08/2025	Prototipo UI login/registro	Aceptar flujo de verificación por correo

4	4/09/2025	PoC graficación (Plotly/Matplotlib)	Elegir Plotly para web; fallback estático
5	11/09/2025	Errores de sintaxis y feedback	Implementar resaltado de posición del error
6	18/09/2025	Historial: búsqueda/etiquetas	Definir paginación y soft delete
7	25/09/2025	Parámetros y estilo de gráfica	Guardar presets por usuario
8	2/10/2025	Exportación PNG/SVG	Límite de resolución + metadatos
9	9/10/2025	Admin activar/desactivar	Registrar auditoría mínima
10	16/10/2025	2FA TOTP	Flujos de recuperación y soporte
11	23/10/2025	Pruebas integración	Métricas TTI; carga en móviles
12	30/10/2025	Demo final interna	Checklist de despliegue y rollback plan

### **Impacto del Proyecto**

El proyecto EcuPlotWeb tiene el potencial de generar impactos en varios niveles: personal (estudiantes, docentes), institucional (programa, universidad) y social (acceso al conocimiento, equidad educativa). A continuación, se describen algunos de los impactos esperados, sustentados con estudios reales.

#### **Impacto en el Aprendizaje y en el Estudiante**

- Mejora en la comprensión de conceptos matemáticos abstractos: Las herramientas digitales y visualización gráfica facilitan que los estudiantes entiendan funciones como las lineales, cuadráticas, etc., al permitir explorar dinámicamente sus gráficas.

Estudios muestran que al usar aplicaciones matemáticas los alumnos tienen una

interacción más significativa con los conceptos, lo que potencia el rendimiento académico en matemáticas.

- Aumento de la motivación y compromiso: Al permitir graficar rápidamente y experimentar con funciones sin la barrera técnica, los estudiantes tienden a sentirse más implicados en su proceso de aprendizaje. Se ha documentado que herramientas digitales despiertan interés, participación activa y mejor actitud hacia temas que solían generar frustración (por ejemplo, funciones lineales).

### **Impacto Institucional y de Gestión del Conocimiento**

- Aumento de la motivación y compromiso: Al permitir graficar rápidamente y experimentar con funciones sin la barrera técnica, los estudiantes tienden a sentirse más implicados en su proceso de aprendizaje. Se ha documentado que herramientas digitales despiertan interés, participación activa y mejor actitud hacia temas que solían generar frustración (por ejemplo, funciones lineales).
- Mejora en la calidad docente y en la innovación educativa: Cuando los docentes cuentan con herramientas accesibles, se facilita la incorporación de actividades más dinámicas en clase, fomentando métodos activos, visuales e interactivos. Esto puede transformar la forma de enseñar, disminuir la carga de explicaciones solo simbólicas, y fortalecer el acompañamiento pedagógico.

### **Impacto Social y en Equidad**

- Accesibilidad educativa más amplia: Al ser una herramienta web gratuita y de acceso sencillo (modo invitado), contribuye a reducir barreras socioeconómicas que impiden que algunos estudiantes accedan a software costoso o complejas instalaciones. Estudios latinoamericanos señalan que una buena parte de la brecha educativa digital depende no solo de acceso a internet sino de la facilidad de uso de las herramientas.

- Fomento del pensamiento lógico y resolución de problemas: El uso de herramientas digitales interactivas se asocia con mejores habilidades de pensamiento crítico, lógico-matemático, ya que obliga a los estudiantes a formular funciones, interpretar sus resultados visuales, diagnosticar errores y ajustar su enfoque.

### **Impacto Esperado Específico de EcuPlotWeb**

Con base en estos estudios, se espera que EcuPlotWeb logre:

1. Que los estudiantes de Fusagasugá mejoren su rendimiento académico en cursos que usan funciones matemáticas, pues podrán practicar más, visualizar más y retomar lo que ya hicieron.
2. Que los docentes tengan menos barreras para incorporar actividades visuales interactivas en sus clases, lo que se traduce en clases más dinámicas y mejor recepción por parte de los estudiantes.
3. Que el programa de Ingeniería de Sistemas y Computación fortalezca su capacidad para gestionar el conocimiento generado por los estudiantes, lo que puede reflejarse en mejores prácticas de enseñanza, documentación institucional y mejora continua.
4. Que exista un aporte concreto hacia la equidad: estudiantes con menos recursos o con menos familiaridad tecnológica puedan usar la herramienta en modo invitado, sin necesidad de hardware especializado ni licencias costosas, disminuyendo diferencias de acceso.

### **Aporte de los Otros Cadi del Semestre**

El desarrollo de EcuPlotWeb no surge de un vacío, sino de la integración de los conocimientos adquiridos en los diferentes Campos de Aprendizaje Disciplinario (CADI) que cursamos en este semestre. Cada CADI aporta elementos específicos que fortalecen tanto el diseño como la implementación y gestión del proyecto.

## Ingeniería de Software II

Este CADI nos da las bases para entender cómo debe gestionarse un proyecto de desarrollo de software. Aquí hemos aprendido la importancia de aplicar metodologías ágiles como XP, organizar las fases de trabajo, documentar adecuadamente los procesos y asegurar la calidad del producto final. Como Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca Teléfono: (601) 8281483 Línea Gratuita: 018000180414 [www.ucundinamarca.edu.co](http://www.ucundinamarca.edu.co) E-mail: [info@ucundinamarca.edu.co](mailto:info@ucundinamarca.edu.co) NIT: 890.680.062-2 Documento controlado por el Sistema de Gestión de la Calidad Asegúrese que corresponde a la última versión consultando el Portal Institucional señala Pressman (2014), la ingeniería de software moderna no solo se centra en programar, sino en planificar, coordinar y mantener proyectos que respondan a necesidades reales, aplicando buenas prácticas y ciclos de mejora continua.

### **Networking**

Desde este campo disciplinar se obtiene la visión práctica de cómo se conectan los equipos, la administración de redes y la transmisión de datos entre dispositivos. Estos conocimientos son esenciales para pensar en la escalabilidad del proyecto y en cómo un software como EcuPlotWeb puede funcionar de manera estable cuando diferentes usuarios lo consultan desde múltiples ubicaciones. Según Tanenbaum y Wetherall (2011), el entendimiento de protocolos y prácticas de red es clave para garantizar servicios confiables y accesibles en entornos distribuidos como la web.

### **Comunicación de Datos**

Este CADI aporta la comprensión de cómo se intercambia información a través de redes IP y cómo se estructura el flujo de datos en sistemas distribuidos. Para nuestro proyecto, estos fundamentos aseguran que las funciones enviadas por el estudiante al sistema (en formato texto) puedan convertirse en datos transmitidos de manera segura y eficiente hasta ser procesados y devueltos en forma de gráfica. Como plantean Stallings y Case (2013), la

comunicación de datos efectiva se convierte en la columna vertebral de cualquier aplicación moderna que dependa de internet.

## Conclusiones

### **La visualización si acerca la matemática**

Pasar del bot de Telegram a una app web con graficas interactivas, cambia todo: el estudiante no solo “ve” la función, la explora (acerca, mueve, lee valores) y eso baja la barrera de lo abstracto. Para el docente, el historial por usuario deja rastro de intento y facilita retroalimentar con ejemplo reales de la clase.

### **La Arquitectura Quedo Lista para Crecer sin Romperse**

Con un diseño modular (cliente web, API, motor de graficacion, correo y 2FA, base de datos en 3FN), el sistema es mantenible y seguro sin volverlo pesado. Exportar (PNG/SVG), reusar funciones del historial y filtrar búsquedas demuestran que se puede ofrecer valor practico sin complicar la vida al usuario.

### **Las Piedras del Camino Están Claras**

Hay retos que no se maquillan: conectividad irregular, dispositivos diversos, expresiones mal escritas o con dominios problemáticos, y cierta curva para activar el 2FA. La salida del diseño es honesta: mensajes de error pedagógicos, rangos por defecto sensatos, tiempos de respuesta cuidados, y opciones claras (invitado vs. registrado) para no frenar a nadie.

### **Lo que Sigue es Validar con Aula Real**

El producto está en un punto bueno para pilotos: medir uso, recoger fricción, cerrar brechas de accesibilidad (teclado, contraste), y priorizar mejoras del backlog (favoritos, etiquetas, micro interacciones). Con un cronograma vivo y pruebas con usuarios, EcuPlot puede pasar de “promesa” a herramienta cotidiana en clase ligera, confiable y hecha para aprender.

## Anexos

### Enlace Github

<https://github.com/RicoCardenas/webapp>

<https://github.com/RicoCardenas/ecuplot>

### Enlace Airtable

[https://airtable.com/invite/l?inviteId=inv6m9sl5koRlkIxY&inviteToken=37f08bc38e3c08ad e82eabf2e04362dd2e11cd48abfa4b58deaf7c6e4eb3e265&utm\\_medium=email&utm\\_source=product\\_team&utm\\_content=transactional-alerts](https://airtable.com/invite/l?inviteId=inv6m9sl5koRlkIxY&inviteToken=37f08bc38e3c08ad e82eabf2e04362dd2e11cd48abfa4b58deaf7c6e4eb3e265&utm_medium=email&utm_source=product_team&utm_content=transactional-alerts)

## Referencias

- Dash. (s. f.). *Interactive graphing and crossfiltering*. <https://dash.plotly.com/interactive-graphing>
- International Telecommunication Union. (2022a). *Global connectivity report 2022*.  
[https://www.itu.int/dms\\_pub/itu-d/opb/ind/d-ind-global.01-2022-pdf-e.pdf](https://www.itu.int/dms_pub/itu-d/opb/ind/d-ind-global.01-2022-pdf-e.pdf)
- International Telecommunication Union. (2022b). *Global connectivity report 2022 – Executive summary*. [https://www.itu.int/dms\\_pub/itu-d/opb/ind/D-IND-GLOBAL.01-2022-SUM-PDF-E.pdf](https://www.itu.int/dms_pub/itu-d/opb/ind/D-IND-GLOBAL.01-2022-SUM-PDF-E.pdf)
- Lowrie, T., Logan, T., & Hegarty, M. (2023). Spatial visualization supports students' math: Mechanisms of effect and transfer. *Journal of Intelligence*, 11(6), 127.  
<https://www.mdpi.com/2079-3200/11/6/127>
- Ministerio de Tecnologías de la Información y las Comunicaciones. (2024, 2 de mayo). *Colombia cerró el año 2023 con 6,1 millones de nuevos accesos a Internet*.  
<https://www.mintic.gov.co/portal/715/w3-article-382041.html>
- Ministerio de Tecnologías de la Información y las Comunicaciones. (2024, 18 de octubre). *Índice de Brecha Digital 2023*. <https://colombiatic.mintic.gov.co/679/w3-article-396961.html>
- Ministerio de Tecnologías de la Información y las Comunicaciones. (2025). *Informe de gestión 2024*. [https://www.mintic.gov.co/portal/715/articles-399819\\_recurso\\_1.pdf](https://www.mintic.gov.co/portal/715/articles-399819_recurso_1.pdf)
- OWASP Foundation. (2024a). *Authentication cheat sheet*.  
[https://cheatsheetseries.owasp.org/cheatsheets/Authentication\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html)
- OWASP Foundation. (2024b). *Multifactor authentication (MFA) cheat sheet*.  
[https://cheatsheetseries.owasp.org/cheatsheets/Multifactor\\_Authentication\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html)
- Plotly. (s. f.). *Plotly Python graphing library*. <https://plotly.com/python/>

Schoenherr, J. (2024). Learning with visualizations helps: A meta-analysis of visualization interventions in mathematics education. *Studies in Educational Evaluation*.

<https://www.sciencedirect.com/science/article/pii/S1747938X24000484>

Schoenherr, J. (2024). Characterizing external visualization in mathematics education research: A scoping review. *ZDM – Mathematics Education*.

<https://link.springer.com/article/10.1007/s11858-023-01494-3>

Schwaber, K., & Sutherland, J. (2020). *The 2020 Scrum Guide*. <https://scrumguides.org/scrum-guide.html>

Scrum.org. (2024). *What is a product backlog?* <https://www.scrum.org/resources/what-is-a-product-backlog>

Matplotlib. (s. f.). *Matplotlib documentation — Matplotlib 3.9.0 documentation*.

<https://matplotlib.org/stable/index.html>

Pro-Ciencia. (2018, 17 de agosto). *Python: Graficar funciones con MATPLOTLIB 1* [Video]. YouTube. <https://www.youtube.com/watch?v=hUfHUPPCxZ4>

Pyrogram. (s. f.). *Welcome to Pyrogram — Pyrogram documentation*. <https://docs.pyrogram.org/>

Telegram. (s. f.). *Telegram Bot API*. <https://core.telegram.org/bots/api>

TeLe TiPs – Telegram. (2021, 13 de octubre). *#0 introduction | Learn Pyrogram | How To Code A Telegram Bot* [Video]. YouTube. <https://www.youtube.com/watch?v=9BtAmuu3RZY>