

K2A Developers:

Aplicación para desarrollo y organización de proyectos
propulsada por IA

Desafío STEAM

TICS Aplicados a la Informática

Proyecto para Expotecnica 2026

Ángel Torres Monge, Kevin Barth Scoth, Kiriath Fernández Bustamante

Ronny Duran Mora

Colegio Técnico Profesional Roberto Gamboa Valverde

CORVEC Unidos por la Excelencia

amgeltormog@gmail.com

barthkevin987@gmail.com

kiriathDariel@gmail.com

Resumen:

K2A Developers es una aplicación desktop que integra un editor de código avanzado, un sistema de organización de proyectos y un asistente de inteligencia artificial ejecutado de forma local. Desarrollada en Python con las librerías PyQt y Monaco Editor, la plataforma busca ofrecer una solución completa, accesible y privada para estudiantes y programadores principiantes o intermedios.

A través de Ollama y el modelo Llama 3.2, la aplicación proporciona sugerencias inteligentes, optimizaciones y explicaciones sin generar código completo, manteniendo al usuario como protagonista de su propio proyecto. Esta implementación local garantiza la privacidad de la información y permite su uso sin conexión a internet constante. El proyecto se enmarca en la categoría **Desafío STEAM – Tecnologías de la Información Aplicadas a la Informática**, con el objetivo de promover herramientas educativas que fomenten el aprendizaje autónomo y la organización eficiente de proyectos de software.

Introducción

Planteamiento del Problema: ¿Existe una aplicación en la actualidad que pueda proporcionar la posibilidad de desarrollar código efectivo, eficiente y optimizado junto con la funcionalidad de poder manejar un proyecto de manera factible y beneficiosa, propulsada por IA?

Antecedentes: Para entender mejor el propósito de K2A Developers, es útil observar otros programas similares que ya existen en el mercado. El referente más conocido es Visual Studio Code (Microsoft, 2026), un entorno muy completo y utilizado a nivel mundial. Por otro lado, han surgido opciones como Cursor IDE, Windsurf y GitHub Copilot, que destacan por incorporar asistencia mediante inteligencia artificial. Lo que hace diferente a K2A Developers es que toma esas buenas ideas y las combina con una gestión de proyectos propia, además de ejecutar la inteligencia artificial directamente en la computadora del usuario, lo que garantiza una mayor privacidad de los datos.

En la actualidad hay muchas aplicaciones que intentan ayudar a los programadores durante su trabajo algunas se enfocan en organizar proyectos otras en escribir código y otras en dar sugerencias mediante inteligencia artificial pero lo que ha sido claro es que casi ninguna junta todo eso en un mismo lugar de una forma que realmente se sienta cómoda y útil para quien la usa esa fue precisamente la pregunta que se realizó al inicio de la introducción.

Cuando se inició la investigación para buscar la respuesta a esa pregunta el equipo se percató de que las opciones que existían o eran demasiado complejas o estaban pensadas para un tipo de usuario muy específico o simplemente no ofrecían la combinación de herramientas que se tenía en mente había editores de código muy buenos pero sin funciones de administración de proyectos había

gestores de proyectos pero sin editor y había herramientas con inteligencia artificial pero que funcionaban de una manera que al equipo no convenía especialmente porque muchas de ellas simplemente le escriben el código al usuario sin explicar nada y eso en nuestra opinión no ayuda a aprender ni a mantener el control sobre el propio trabajo.

Fue a partir de eso que nació la idea de desarrollar K2A Studio una aplicación pensada para programadores que quieren tener todo en un mismo lugar sin complicaciones la idea no era crear algo exagerado ni lleno de funciones que nadie usa sino algo práctico que resolviera un problema real que el equipo había sentido en algún momento durante el desarrollo de proyectos.

Una de las decisiones más importantes que se tomó desde el principio fue definir como funcionara la inteligencia artificial dentro de la aplicación porque ese era un punto donde se podía caer fácilmente en hacer lo mismo que ya hacen otras herramientas se decidió que no se quería una inteligencia artificial que simplemente generara código completo y se lo entregara al usuario porque aunque eso puede parecer una ventaja en realidad tiene un problema bastante serio cuando alguien recibe código que no escribió el mismo muchas veces no entiende cómo funciona no sabe por qué está escrito de esa forma y si algo falla no tiene idea de cómo resolverlo eso hace que el programador pierda el hilo de su propio proyecto y empieza a depender de la herramienta en vez de desarrollar su propio criterio

Por eso el equipo resolvió optar por un enfoque diferente la inteligencia artificial dentro de k2a studio está pensada para acompañar al usuario no para reemplazarlo puede revisar el código que el usuario ya tiene escrito identificar partes que podrían mejorarse sugerir una organización más limpia o explicar que hace una sección determinada pero siempre de una forma que invite al usuario a entender y decidir por sí mismo nunca imponiendo cambios ni sobrescribiendo lo que ya está hecho

Además el equipo decidió que la aplicación fuera algo que cualquier persona pudiera usar sin necesitar una computadora de último modelo ni una conexión a internet constante por eso gran parte de sus funciones incluyendo la inteligencia artificial corren de manera local directamente en el equipo del usuario eso también significa que la información de los proyectos no sale del computador en ningún momento lo cual al equipo le pareció fundamental desde el principio considerando que muchos programadores trabajan con código que contiene información privada o sensible

En resumen, K2A Studio nace como una respuesta a algo que el equipo sintió que faltaba una herramienta que combine la edición de código la administración de proyectos y el apoyo de inteligencia artificial en un solo lugar de una manera que sea útil accesible y que sobre todo mantenga al usuario en control de su propio trabajo en todo momento.

Metodología:

La aplicación fue desarrollada por medio de Python, css y html en VS Code y fue almacenada en la nube en GitHub. Los colaboradores fueron Ángel David Torres Monge, Kiriath Dariel Fernández Bustamante y Kevin Josué Barth Scoth. El trabajo inicio el 14 de mayo de 2026 a el 25 de junio de 2026. Se desarrollo en el CTP Roberto Gamboa Valverde.

1. Problemática de programación:

Se nos complicó implementar el editor de texto Monaco Editor porque este funciona en la web y no trabaja directamente con Python. Por eso no solo se utilizó Python en el proyecto, sino que también fue necesario trabajar con HTML, JavaScript y CSS para que todo funcionara correctamente dentro de la aplicación. Coordinar estos lenguajes con Python se complicó, principalmente porque no se contaba con mucha experiencia con la integración de tecnologías web dentro de una aplicación de escritorio.

2. Investigación que desarrollamos:

Durante la investigación se aprendió a usar Python como base para el desarrollo de la aplicación y también se investigó como usar herramientas como PyQt5 para crear la interfaz y PyQtWebEngine para integrar contenido web dentro del programa. También se vio cómo funciona el editor Monaco y cómo puede trabajar con HTML, CSS y JavaScript para adaptarlo al proyecto. Además, se exploraron módulos como os y json para manejar archivos y guardar información, y se analizó la implementación de inteligencia artificial usando Ollama para dar sugerencias dentro de la aplicación.

3. Implementación y desarrollo de app y la vinculación de la IA:

El desarrollo de la aplicación se realizará de forma progresiva utilizando Visual Studio Code como entorno principal de desarrollo. Inicialmente se implementarán las funcionalidades básicas del sistema y posteriormente se incorporarán herramientas que permitan mejorar la asistencia al usuario durante el proceso de programación. Cada módulo será probado de manera individual para verificar su correcto funcionamiento y asegurar la estabilidad general de la aplicación.

La vinculación de la inteligencia artificial se llevará a cabo mediante Ollama utilizando el modelo Llama 3.2 ejecutado localmente. La aplicación establecerá una comunicación con el modelo para enviar consultas relacionadas con el código o el proyecto en desarrollo. A partir de esta información, la IA proporcionará sugerencias, posibles correcciones y explicaciones que servirán como apoyo al usuario. Esta integración permitirá aprovechar las capacidades de la inteligencia artificial manteniendo un procesamiento local y una mayor privacidad de los datos utilizados durante el desarrollo.

4. Ejecución de la app:

La aplicación será convertida momentáneamente en un archivo de formato .exe, aunque esta aplicación es capaz de ejecutarse en Linux y Mac os, el equipo desea primero probar en Windows y luego convertirla para su uso en los otros sistemas operativos. Así cualquier persona solo descarga un archivo

ejecutable e inicia la aplicación. En el futuro se desea desarrollar una aplicación aparte o una extensión más completa para instalar la aplicación y permitir que esta no sea un solo archivo con formato .exe.

5. Intentar publicarlo como parte del negocio:

La aplicación actualmente se encuentra en GitHub, un repositorio online muy conocido y con alta reputación. Por medio de este repositorio el equipo optara por hostear una página web donde cualquier usuario en internet pueda encontrarla de forma segura y sencilla. Esto está planeado para el futuro debido a que la aplicación no está terminada.

Se trabajó bajo el siguiente cronograma:

Actividad	Responsables	Fechas	Estado
Desarrollo de la base del programa e investigación de las herramientas que se usaran	Angel David Torres Monje Kiriath Dariel Fernández Bustamante Kevin Josué Barth Scoth	7/5/2026	completo
Mejora visual y del programa al igual que la investigación de la implementación de mejoras visuales	Ángel David Torres Monje Kevin Josué Barth Scoth	21/5/2026	completo
Investigación de como implementar los módulos al programa	Ángel David Torres Monje	28/5/26	completo
Implementación de los modulo al programa	Ángel David Torres Monje	4/6/2026	completo
Investigación de como implementar la AI al programa	Kiriath Dariel Fernández Bustamante	11/6/26	pendiente
Investigación de como entrenar a al AI para las funciones necesarias	Kiriath Dariel Fernández Bustamante	25/6/2026	pendiente

Marco teórico:

Para desarrollar una aplicación por la cual se pueda manejar proyectos se decidió aplicar el conocimiento del lenguaje de programación de Python:

Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos.

Por medio de Python se pueden usar módulos capaces de facilitar el trabajo, se investigaron distintos módulos que se podían usar adentro de Python para programar, sin embargo, el equipo estaba más familiarizado con uno en concreto llamado PyQt5

PyQt es un binding de la biblioteca gráfica Qt para el lenguaje de programación Python. Específicamente se encarga de proveer como antes dicho un set de herramientas para la facilitación de la implementación de interfaces de usuario dentro de la aplicación.

A partir de esto se desarrolló la interfaz de usuario implementando la librería anteriormente mencionada. Se desarrollo un archivo principal donde se podían acceder a diversas herramientas propuestas por la librería. El uso de esta herramienta necesita de conocimiento previo, si no se tiene accedemos a la página oficial de la documentación de PyQt.

La aplicación usa otros módulos como:

os: Modulo que permite la interacción con el sistema operativo, su función más importante es permitir que la aplicación se comunique con el sistema de archivos y se adapte al sistema operativo del usuario para evitar errores. Nuevamente, si se desconoce de información necesaria para usar este módulo se ingresó a la página.

json: Modulo que permite codificar objetos de python y decodificarlos, se usa en el proyecto para crear archivos donde se puede almacenar datos en formato JSON y así guardar información importante para que la aplicación funcione de forma efectiva.

La aplicación posee una característica importante que es la posibilidad de editar el proyecto del usuario como en un IDE que es un Entorno de Desarrollo Integrado donde se combinan herramientas esenciales para programadores, como un editor de código: Se busco diversos editores de código, puesto que desarrollar uno propio es algo poco rentable para la escala de este proyecto.

Entre los frameworks más rentables para el proyecto se encontraron dos:

CodeMirror: Es un editor de código para la web. Puede usarse en sitios web para implementar el código con soporte a muchas características, con una rica interfaz de programación.

Monaco Editor: Es el editor de código que está en Visual Studio Code. Al igual que CodeMirror tiene bastantes características y es un editor que corre en la web.

Se opto por usar Monaco editor porque este es mayormente usado por aplicaciones IDE como Visual Studio Code, Cursor, Windsurf.

Para implementar Monaco editor en la aplicación, se indaga en la documentación oficial de PyQt donde al conocer cómo es que el editor funciona (funciona en la web). Se encontró un módulo extra que no viene instalado por defecto en el módulo de PyQt, se llama PyQtWebEngine.

PyQtWebEngine es un framework que provee la habilidad de crear aplicaciones basadas en la web y en el navegador

Chrome. Permitiendo integrar páginas web con HTML, CSS y JavaScript en la aplicación desarrollada con Python, además, permite la comunicación entre la aplicación y la página web.

Para implementar características propias de la aplicación como el sistema de “Tabs” o “Pestañas” y el chat del asistente, estos se desarrollaron desde cero. Aunque las otras características si tienen una base que viene directamente desde la librería de PyQt.

A la hora de meterle inteligencia artificial a K2A Developers, el equipo no quería depender de las típicas opciones de siempre basadas en la nube que consumen datos o que obligan a los usuarios a estar conectados a internet a la fuerza. Por eso, el grupo de desarrollo se la jugó por Ollama, una plataforma espectacular que permite correr modelos de lenguaje de forma 100% local directo en la máquina del programador. Entre todo el abanico de opciones disponibles en el sector, se seleccionó específicamente Llama 3.2, que es una versión súper ligera, optimizada y eficiente que sacó la gente de Meta. Lo mejor de esta IA es que vuela sin necesidad de colgarse a una red y está pensada para adaptarse sin problemas a computadoras convencionales, evitando que se trabe el sistema o se sobrecaliente el equipo mientras se trabaja. Esta movida técnica no fue una casualidad, sino que respondió a la obsesión del equipo por blindar la privacidad del código y de los proyectos. Como todo el procesamiento de datos se queda guardado dentro del mismo dispositivo, se elimina de raíz el miedo a que información sensible o ideas clave terminen filtrándose en servidores externos o bases de datos de terceros.

Además, el enfoque que se le dio a este asistente es totalmente diferente al de otras herramientas del mercado que autocompilan bloques enteros de código de golpe, algo que a la larga solo hace que las personas trabajen en piloto automático, no entendiendo lo que hacen y se vuelvan dependientes. En K2A Developers, la IA se diseñó desde cero para ser un verdadero compañero de equipo que trabaja codo a codo con el usuario. Aprovechando el potencial de comprensión de Llama 3.2, el asistente se convierte en una especie de mentor técnico capaz de revisar el código que el usuario ya lleva avanzado, tirando ideas geniales para limpiar, refactorizar y ordenar la estructura general de los archivos. También funciona de maravilla como un cazador de errores en tiempo real, detectando fallas de sintaxis o de lógica mucho antes de que el código se rompa al correrlo, y tiene la facilidad de explicar de forma súper clara esas secciones pesadas o algoritmos confusos que a veces cuestan entender a primera vista. Con esto se logra que los desarrolladores nunca pierdan las riendas de su software, entiendan el porqué de cada sugerencia y tomen la última palabra, impulsando al máximo el aprendizaje diario y la autonomía frente a la pantalla.

Para que toda esta combinación funcione a nivel interno de forma fluida y sin retrasos molestos, la integración se logró conectando la API local de Ollama directamente con el módulo de chat que el usuario tiene a mano en la interfaz principal de la app. Esta conexión directa es clave porque le da superpoderes a la IA para entender el contexto completo de lo que se está haciendo al instante, leyendo desde los archivos que están abiertos en las pestañas hasta la jerarquía de las carpetas y las líneas de código existentes en ese preciso momento. Gracias a esto, el asistente no tira respuestas genéricas de internet, sino soluciones ultra rápidas,

personalizadas y útiles que van directo al grano de lo que se necesita resolver en el código. Al final del día, armar esta infraestructura local no solo garantiza que la aplicación sirva perfectamente a mitad del campo o si se cae el internet de la casa, sino que encaja de lleno con el verdadero ADN de K2A Developers, que no es crear una IA floja que programe por la gente, sino diseñar una herramienta moderna que potencie, acompañe y suba el nivel de cada desarrollador sin intentar reemplazar su talento.

Para crear la interfaz de nuestra aplicación K2A Developers lo primero que se hizo fue definir como queríamos que se viera la página y como queríamos que el usuario se sintiera al abrir la aplicación por primera vez. Así que lo que decidimos hacer fue separar todo por partes bien definidas para que nadie tuviera que adivinar donde esta cada cosa. Se eligieron colores para algo en específico no solo porque la aplicación se viera bien, usamos tipografías más simples porque al ser una aplicación de programación maneja bastante contenido técnico y necesitábamos que todo fuera fácil de leer.

También pensamos mucho en cómo se distribuían los elementos dentro de la aplicación. Queríamos que las herramientas mas importantes estuvieran siempre al alcance del usuario para evitar que tuviera que buscar mucho por los menús para realizar tareas simples. Por eso se organizaron diferentes funciones en secciones específicas, permitiendo meterse fácilmente al editor de código, al sistema de administración de proyectos y al asistente de inteligencia artificial.

Otra parte importante del diseño fue buscar que la interfaz se sintiera familiar para las personas que ya utilizan otras apps de programación. Para eso nos inspiráramos en aplicaciones modernas utilizadas por desarrolladores, manteniendo una estructura ordenada y fácil de entender. También desarrollamos nuestro propio sistema de pestañas para que el usuario pudiera trabajar con varios archivos al mismo tiempo de una manera más cómoda.

Además, se realizaron distintos ajustes para mejorar la experiencia visual y hacer que la aplicación se sintiera más profesional. Se cuidaron detalles como el tamaño de los textos, la distribución de los botones y la organización de cada ventana. El objetivo principal siempre fue que el usuario pudiera concentrarse en desarrollar y administrar sus proyectos sin distraerse con una interfaz complicada o desordenada.

También se consideró importante que el diseño estuviera relacionado con las funciones de la aplicación. No solo se buscó que la interfaz se viera bien, sino que cada cosa que estuviera dentro de la plataforma tuviera un propósito. Esto ayuda a que el usuario pueda notar más rápido para que sirve cada herramienta y también entender mejor como usar las diferentes opciones que da la aplicación.

Y también se tomó en cuenta que una persona puede pasar mucho tiempo utilizando la aplicación mientras trabaja en algún proyecto. Por eso, se buscó que el uso de la aplicación se sintiera más natural y cómoda durante todo el proceso. La idea fue que el usuario pudiera cambiar entre diferentes tareas sin sentir que el trabajo se vuelve pesado o complicado. También se intentó que el paso entre

una sección y otra fuera sencillo para evitar interrupciones innecesarias mientras se desarrolla un proyecto.

Objetivos

Objetivo General: Implementar una plataforma de programación enfocada en la revisión de código fuente de software variado mediante el uso combinado de inteligencia artificial para la optimización de proyectos.

Objetivos Específicos:

Desarrollar una plataforma de revisión de código fuente y de manejo de proyectos utilizando distintos lenguajes de programación para administrar los proyectos de los programadores

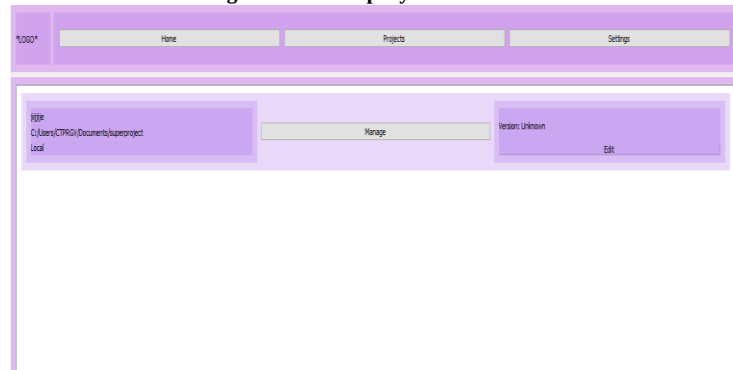
1. Implementar el enlace con IA mediante un protocolo que permite a nuestra plataforma para comunicarse con la IA por medio de herramientas

Resultados y Conclusiones:

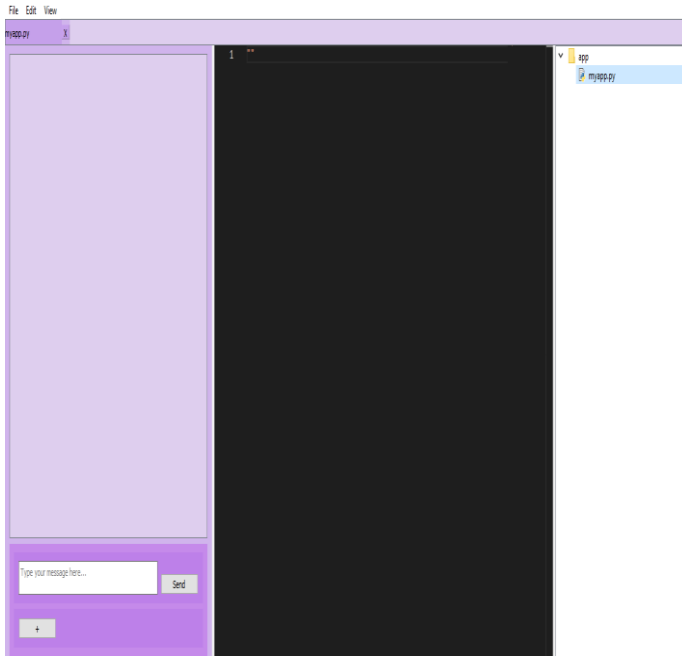
Página de inicio:



Página de lista de proyectos



Página del editor de código:



Se contaba con la duda de decidir entre cual librería o framework usar para la interfaz de usuario, debido a que el equipo poseía más experiencia en python; se optó por usar PyQt5 como framework para reducir la posibilidad de problemas futuros.

A la hora de implementar la funcionalidad principal que se basa en poder editar un proyecto como un IDE, surgieron problemas para lograr comunicar el código con la aplicación; el editor trabaja con html y css. Conectar ambos fue un problema en el que se necesitó de investigación en la página oficial puesto que, el equipo no contaba con experiencia para conectar este editor.

Cuando se insertó el sistema de chat con el asistente aparecieron problemas a la hora de crear visualmente el chat, debido a que la estructura es compleja y tiene que ser muy ordenada para que los mensajes se acomoden de forma correcta y se adapten al tamaño de la pantalla, fue complicado agregar esta característica porque al probar los mensajes algunos no aparecían correctamente. Nuevamente se logró solventar e incluso agregar escalamiento a la

característica por medio de una investigación y experiencia de nuestro equipo.

Concluyendo, desarrollar el prototipo de la aplicación fue algo tardado y surgieron algunos desafíos, sin embargo, fueron medianamente sencillos de solventar a la hora de investigar una solución efectiva.

Referencias

- CodeMirror Project. (s. f.). CodeMirror. <https://codemirror.net/>
- Interaction Design Foundation. (s. f.). *User interface (UI) design*. <https://www.interaction-design.org/literature/topics/ui-design> Recuperado el 11 de junio de 2026, de [URL]
- Material Design. (s. f.). *Material Design 3*. Google. <https://m3.material.io/> Recuperado el 11 de junio de 2026, de [URL]
- Meta. (2024). *Llama 3.2*. <https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/>
- Microsoft. (2026). *Visual Studio Code* (versión 1.102) [Software]. <https://code.visualstudio.com>
- Microsoft. (s.f.). *Monaco Editor*. Recuperado el 11 de junio de 2026, de <https://microsoft.github.io/monaco-editor/>
- Nielsen Norman Group. (s. f.). *User interface design guidelines*. <https://www.nngroup.com/articles/user-interface-design-guidelines/> Recuperado el 11 de junio de 2026, de [URL]
- Ollama. (s.f.). *Ollama*. Recuperado el 11 de junio de 2026, de <https://ollama.com/>
- Python Software Foundation. (s. f.). json — JSON encoder and decoder. Documentación oficial de Python. <https://docs.python.org/es/3.9/library/json.html>
- Python Software Foundation. (s. f.). os — Miscellaneous operating system interfaces. Documentación oficial de Python. <https://docs.python.org/3/library/os.html>
- Python Software Foundation. (s.f.). *Python 3 Documentation*. Recuperado de <https://docs.python.org/es/3/>
- Python Software Foundation. (s. f.). Tutorial de Python. Documentación oficial de Python. <https://docs.python.org/es/3/tutorial/>
- Riverbank Computing. (s.f.). *PyQt5 Reference Guide*. Recuperado de <https://www.riverbankcomputing.com/static/Docs/PyQt5/>
- Wikipedia contributors. (s. f.). Integrated development environment. Wikipedia. https://en.wikipedia.org/wiki/Integrated_development_environment
- Wikipedia contributors. (s. f.). PyQt. Wikipedia. <https://en.wikipedia.org/wiki/PyQt>

Colegio Técnico Profesional

Roberto Gamboa Valverde

CORVEC Desamparados

**Escudo del
centro
educativo**

Bitácora para los proyectos de ExpoTÉCNICA

Bitácora N° 01

Fecha: 7/5/26

Nombre completo de las personas estudiantes: Kiriath daniel Fernandez bustamante,
Angel Torres Monge, Kevin Barth Scoth

Hora inicio: 12:15

Hora finalización: 2:10

Materiales utilizados: equipo de cómputo y sistemas móviles

Lugar: CTP Roberto Gamboa Valverde

1. Actividad (es) realizada

(s): 1.1 investigation

1.2 elaboración del trabajo escrito

2. Producto de la actividad realizada (datos, información, resultados de experimentos, entre otros):

2.1 trabajo escrito

2.2 trabajo escrito

3. Análisis de los productos recopilados:

El uso de los MCP puede llegar a facilitar nuestro trabajo funcionando como un puente entre los sistemas de inteligencia artificial y el software desarrollado al igual que no hay ningún programa desarrollado que compile todo lo que nuestra idea quiere desarrollar

4. Acciones requeridas:

Acción	Responsable	Fecha de entrega	Estado
Investigación sobre nuestra idea	Kiriath, Angel y Kevin	7/5/2026	Pendiente
Elaboración del trabajo escrito	Kiriath, Angel y Kevin	7/5/2026	Pendiente

Observaciones:

Firma de estudiantes:

Angel David Torres Monge Kevin Josue Barth scoth
Kiriath Darriel Fernandez Bustamante

Firma de la persona tutora

Colegio Técnico Profesional

Roberto Gamboa Valverde

CORVEC Desamparados

**Escudo del
centro
educativo**

Bitácora para los proyectos de ExpoTÉCNICA

Bitácora N° 2

Fecha: 21/5/26

Nombre completo de las personas estudiantes: Angel Torres Monge, Kevin Barth Scoth, Kiriath Fernández Bustamante

Hora inicio: 12:10 Hora finalización: 4:20

Materiales utilizados: equipo de cómputo y sistemas móviles

Lugar: CTP Roberto Gamboa Valverde

1. Actividad (es) realizada

(s):

1.1 Investigación

1.2 Elaboración del marco teórico del trabajo escrito

1.3 Elaboración de los objetivos

2. Producto de la actividad realizada (datos, información, resultados de experimentos, entre otros):

2.1 Información sobre el marco teórico

2.2 Marco Teórico

2.3 Objetivos

3. Análisis de los productos recopilados:

Se usa Python y PyQt5 para avanzar el desarrollo de la aplicación y su interfaz de usuario como medida preferida.

Nuestros objetivos se basan en implementar una plataforma de programación que venga de la mano con la IA.

4. Acciones requeridas:

Acción	Responsable	Fecha de entrega	Estado
Investigación de nuestras tecnologías utilizadas	Angel Torres Monge	21/5/26	En proceso
Elaboración del marco teórico	Kevin Barth Scoth y Angel Torres Monge	21/5/26	En proceso
Elaboración de los objetivos	Kevin Barth Scoth	21/5/26	En proceso

Observaciones:

Firma de estudiantes:

Angel David Torres Monge Kevin Josue Barth scoth
Kiriath Darriel Fernandez Bustamante

Firma de la persona tutora

Colegio Técnico Profesional

Roberto Gamboa Valverde

CORVEC Desamparados

**Escudo del
centro
educativo**

Bitácora para los proyectos de ExpoTÉCNICA

Bitácora N° 3

Fecha: 28/5/26

Nombre completo de las personas estudiantes: Angel Torres Monge, Kevin Barth Scoth, Kiriath Fernández Bustamante

Hora inicio: 12:10 Hora finalización: 4:20

Materiales utilizados: equipo de cómputo y sistemas móviles

Lugar: CTP Roberto Gamboa Valverde

1. Actividad (es) realizada

(s):

1.1 Canvas

1.2 Introducción trabajo escrito

2. Producto de la actividad realizada (datos, información, resultados de experimentos, entre otros):

2.1 Canvas

2.2 Introducción

3. Análisis de los productos recopilados:

Se creo parte del canvas sobre todo la parte de los socios. Se creo la introducción con la pregunta de cuál aplicación actual solventa el problema que nosotros somos capaces de.

4. Acciones requeridas:

Acción	Responsable	Fecha de entrega	Estado
Introducción	Angel Torres Monge	28/5/26	En proceso
Canvas	Kevin Barth Scoth y Kiriath Fernandez Bustamante	28/5/26	En proceso

Observaciones:

Firma de estudiantes:

Angel David Torres Monge Kevin Josue Barth scoth
Kiriath Darriel Fernandez Bustamante

Firma de la persona tutora

Colegio Técnico Profesional

Roberto Gamboa Valverde

CORVEC Desamparados

**Escudo del
centro
educativo**

Bitácora para los proyectos de ExpoTÉCNICA

Bitácora N° 04

Fecha: 4/6/26

Nombre completo de las personas estudiantes: Kiriath daniel Fernandez bustamante,
Angel Torres Monge, Kevin Barth Scoth

Hora inicio: 12:20

Hora finalización: 4:20

Materiales utilizados: equipo de cómputo y sistemas móviles

Lugar: CTP Roberto Gamboa Valverde

1. Actividad (es) realizada

(s): 1.1 redacción de la
introducción

1.2 elaboración del marco teórico

**2. Producto de la actividad realizada (datos, información, resultados
de experimentos, entre otros):**

2.1 introducción

2.2 avance en la elaboración del marco teórico

3. Análisis de los productos recopilados:

Elaboramos la introducción junto el avance del marco teórico del proyecto

4. Acciones requeridas:

Acción	Responsable	Fecha de entrega	Estado
Elaboración de la introducción	Ángel y Kevin	4/6/2026	Terminado
Avance en el marco teórico	Kiriath, Angel y Kevin	4/6/2026	Pendiente

Observaciones:

Firma de estudiantes:

Angel David Torres Monge Kevin Josue Barth scoth
Kiriath Darriel Fernandez Bustamante

Firma de la persona tutora

Colegio Técnico Profesional

Roberto Gamboa Valverde

CORVEC Desamparados

**Escudo del
centro
educativo**

Bitácora para los proyectos de ExpoTÉCNICA

Bitácora N° 5

Fecha: 11/6/26

Nombre completo de las personas estudiantes: Angel Torres Monge, Kevin Barth Scoth, Kiriath Fernández Bustamante

Hora inicio: 12:10 Hora finalización: 4:20

Materiales utilizados: equipo de cómputo y sistemas móviles

Lugar: CTP Roberto Gamboa Valverde

1. Actividad (es) realizada

(s):

1.1 Trabajo escrito

**1.2 Capturas de
prototipo**

2. Producto de la actividad realizada (datos, información, resultados de experimentos, entre otros):

2.1 Marco Teórico, Resumen

2.2 Capturas prototipo

3. Análisis de los productos recopilados:

Se realizo el resumen después de la portada, se corrigieron problemas con el trabajo escrito y se tomaron capturas de partes específicas del prototipo, además, se completó mas el marco teórico.

4. Acciones requeridas:

Acción	Responsable	Fecha de entrega	Estado
Trabajo escrito	Kevin Barth Scoth y Kiriath Fernandez Bustamante	11/6/26	En proceso
Capturas del prototipo	Angel Torres Monge	11/6/26	Terminado

Observaciones:

Firma de estudiantes:

Angel David Torres Monge Kevin Josue Barth scoth
Kiriath Darriel Fernandez Bustamante

Firma de la persona tutora

Colegio Técnico Profesional

Roberto Gamboa Valverde

CORVEC Desamparados

**Escudo del
centro
educativo**

Bitácora para los proyectos de ExpoTÉCNICA

Bitácora N° 06

Fecha: 25/6/26

Nombre completo de las personas estudiantes: Kiriath Daríel Fernández Bustamante, Ángel David Torres Monge, Kevin Josué Barth Scoth

Hora inicio: 12:15

Hora finalización: 4:20

Materiales utilizados: Equipo de cómputo y sistemas móviles.

Lugar: CTP Roberto Gamboa Valverde

1. Actividad (es) realizada

(s): 1.1. Desarrollo del trabajo escrito del proyecto.

2. Producto de la actividad realizada (datos, información, resultados de experimentos, entre otros):

2.1. Finalización del trabajo escrito del proyecto.

3. Análisis de los productos recopilados:

Se desarrolló el Resumen, la Metodología, se organizó el orden de las partes del trabajo escrito debido a que estaban desordenadas y se trabajó y finalizaron los Resultados y Conclusiones.

4. Acciones requeridas:

Acción	Responsable	Fecha de entrega	Estado
Desarrollo del trabajo escrito del proyecto.	Kiriath Dariel Fernandez Bustamante, Angel David Torres Monge y Kevin Jouse Barth Scoth	25/6/2026	Finalizado

Observaciones:

Firma de estudiantes:

Angel David Torres Monge Kevin Josue Barth scoth
Kiriath Dariel Fernandez Bustamante

Firma de la persona tutora