



## Potencia el desarrollo de software en la industria tecnológica con ideas innovadoras utilizando LLMs

Únete a la primera hackathon conjunta de la Universidad de los Andes y la Universidad del Rosario y descubre cómo marcar la diferencia en el mundo del desarrollo con CodeSavant.

Nombre del proyecto	TestBuddy
Nombre del equipo	Los Buddies
Líder del equipo	Camilo Salinas
Universidad	Universidad de los Andes
Correo electrónico (líder)	ca.salinas@uniandes.edu.co
Teléfono de contacto (líder)	+57 3115904947
Tallas de camisetas de los 5 miembros del equipo	Mujeres: M, M, M Hombres: XL, S

Por favor, responda las siguientes preguntas en relación a su propuesta:

### 1. ¿La propuesta está orientada con el objetivo planteado?

Describe detalladamente cómo su propuesta se alinea con el objetivo del evento de potencializar la realidad empresarial de la industria del desarrollo de software utilizando LLM's.

**TestBuddy** es una extensión basada en Large Language Models (LLMs) que tiene como objetivo optimizar el desarrollo de software mediante la generación automatizada de pruebas unitarias. La propuesta potencializa la realidad empresarial de la industria del desarrollo de software utilizando LLMs.

Algunas características consideradas clave de *TestBuddy* son:

- **Eficiencia y rentabilidad:** *TestBuddy* aborda el impacto en el tiempo y dinero para las empresas al no testear adecuadamente el código. Al automatizar la generación de pruebas unitarias mientras se escribe el código, se reducen problemas y retrasos en el ciclo de desarrollo, lo que ahorra recursos y acelera el proceso.
- **Automatización:** La extensión ofrece una solución automatizada para crear pruebas unitarias de manera temprana y continua. Esto mejora la eficiencia del equipo de desarrollo y permite a los desarrolladores centrarse en tareas más complejas y creativas.
- **Impacto en el Desarrollo de Software:** *TestBuddy* aborda una necesidad crítica en el desarrollo de software al proporcionar pruebas oportunas y precisas. Esto ayuda a reducir el "technical debt" y mejora la calidad del código.
- **Cultura de Desarrollo impulsada por Pruebas:** La extensión fomenta una cultura de desarrollo basada en pruebas (TDD) al integrar la generación de pruebas en el proceso de escritura de código. Esto mejora la calidad del software y reduce errores en el producto final.
- **Solución Integral para fallos en Unit Tests:** *TestBuddy* ofrece una solución All in One Place para abordar fallos en pruebas unitarias. Al crear pruebas en el mismo entorno de desarrollo, se agiliza la detección y corrección de errores.

En pocas palabras, *TestBuddy* es una propuesta innovadora y relevante que aprovecha las capacidades de los LLMs para mejorar la realidad empresarial de la industria del desarrollo de software. Automatizando la creación de pruebas unitarias y promoviendo prácticas de desarrollo basadas en pruebas, *TestBuddy* contribuye a **una mejora significativa en la eficiencia, calidad y rentabilidad de los proyectos de software**.

## 2. ¿Resuelve un problema actual del desarrollo de software en la industria de tecnología?

En el desarrollo de software, la creación manual de pruebas unitarias resulta tediosa y consume tiempo significativo del equipo de desarrollo. La falta de eficiencia y automatización en el proceso de generación de pruebas dificulta la capacidad de los desarrolladores para enfocarse en tareas más complejas y puede generar retrasos en el ciclo de desarrollo, aumentando el riesgo de errores no detectados. Si bien la orientación a pruebas (test-driven) puede ser valiosa y rentable a medida que las compañías crecen y estabilizan sus productos, en la etapa inicial, a menudo se prioriza la velocidad y la entrega rápida para probar sus conceptos en el mercado y obtener tracción.

*TestBuddy* es una propuesta que **contribuye a la solución de este problema al garantizar la calidad del software en un entorno de desarrollo ágil** mediante el uso de Large Language Models (LLMs) para generar pruebas unitarias mientras se escribe el código.

La aplicación de LLMs en *TestBuddy* permite comprender el contexto y la semántica del código en desarrollo, lo que facilita la creación de pruebas unitarias coherentes y precisas. Estos modelos de lenguaje avanzados pueden analizar patrones y detectar problemas potenciales en el código, lo que lleva a una cobertura más completa de pruebas y, en última instancia, a un software más confiable y de alta calidad.

## 3. ¿Qué problema busca resolver? ¿Optimización de tiempo, recursos, entre otros?

Detalle el problema concreto que su propuesta busca resolver y qué aspectos específicos de la optimización (tiempo, recursos, eficiencia, calidad, entre otros) aborda.

**TestBuddy** enfrenta el desafío de productividad y eficiencia durante el desarrollo de software al automatizar la creación de pruebas unitarias. A través de esta solución, se abordan aspectos específicos de optimización, incluyendo tiempo, recursos y calidad del código:

- **Optimización de Tiempo:** Al generar pruebas unitarias de manera automática mientras se escribe el código, *TestBuddy* ahorra tiempo valioso para los desarrolladores. Esta automatización agiliza el proceso de desarrollo y permite una entrega más rápida del software.
- **Optimización de Recursos:** La generación automatizada de pruebas unitarias libera a los desarrolladores de la tarea manual de escribir pruebas, lo que les permite centrarse en tareas más complejas y creativas. Esto optimiza el uso de recursos y aumenta la productividad del equipo de desarrollo.
- **Optimización de Eficiencia:** *TestBuddy* mejora la eficiencia general del proceso de desarrollo al proporcionar pruebas oportunas y precisas. La automatización garantiza una mayor cobertura de pruebas y permite identificar errores tempranamente, lo que reduce el tiempo dedicado al debugging y mejora la eficiencia del equipo.
- **Optimización de Calidad:** Al crear pruebas unitarias de manera automatizada, *TestBuddy* mejora la calidad del código. Las pruebas más completas y confiables contribuyen a un software más robusto y menos propenso a errores, lo que aumenta la satisfacción del cliente y la reputación de la empresa.

#### 4. ¿Es aplicable a un caso de uso en la industria tecnológica?

Describe cómo su propuesta puede ser aplicada en un caso de uso real dentro de la industria tecnológica, brindando ejemplos o escenarios en los que se podría implementar.

En la industria del desarrollo de software, contar con un sistema que permita la automatización de pruebas unitarias es esencial para garantizar una validación rápida y precisa del producto, manejando la eficiencia del desarrollo y asegurando la calidad del mismo. A continuación se presentan diferentes escenarios en los cuales una compañía necesitaría hacer uso de este sistema:

1. **Entrega continua (Continuous Delivery):** Cuando una compañía busca implementar prácticas de entrega continua, el proceso de desarrollo y despliegue de software se vuelve más ágil y rápido. Un sistema de pruebas automatizado garantiza que cada cambio de código se someta a pruebas exhaustivas antes de integrarse con el código base. Esto reduce la probabilidad de errores y fallos en producción, permitiendo una entrega más confiable y frecuente de nuevas características y actualizaciones.
2. **Mantenimiento eficiente:** En compañías con software en funcionamiento, las actualizaciones y correcciones de errores son constantes. Un sistema de pruebas automatizado facilita la identificación temprana de problemas, lo que ahorra tiempo y esfuerzo para el equipo de desarrollo. Al ejecutar pruebas automáticamente en cada iteración del desarrollo, se puede asegurar que las funcionalidades existentes no se vean afectadas negativamente por los cambios realizados.
3. **Colaboración en equipos grandes:** En proyectos donde múltiples equipos trabajan en diferentes componentes del software, las pruebas manuales pueden resultar lentas y propensas a errores humanos. Un sistema automatizado de pruebas unitarias brinda una base común de pruebas, permitiendo que todos los equipos validen sus contribuciones de manera coherente y efectiva.

4. **Reducción de costos y tiempos de desarrollo:** Si bien implementar un sistema de pruebas automatizado puede requerir una inversión inicial, a largo plazo, puede ahorrar tiempo y recursos significativos. Al detectar y corregir errores de manera más rápida y precisa, se reducen los costos asociados con el mantenimiento y los retrabajos, y el desarrollo progresa de manera más eficiente.
5. **Garantizar calidad y confiabilidad:** En la industria del software, la seguridad y la calidad son primordiales y por esta razón un sistema de pruebas automatizado es fundamental. Estas pruebas garantizan que el software cumpla con los estándares de calidad requeridos y que funcione según lo previsto, evitando posibles consecuencias graves en caso de fallos.

**Adicionalmente, incluya:**

Proporcione un resumen (*abstract*) de su proyecto, que incluya una breve descripción de la solución propuesta y su enfoque de implementación. (*Máximo 300 palabras*)

*TestBuddy* es una extensión basada en Modelos de Lenguaje Avanzados (LLMs) que revoluciona la eficiencia en el desarrollo de software a través de la generación automatizada de pruebas unitarias. Su objetivo es potenciar el entorno empresarial de la industria del desarrollo de software mediante la aplicación innovadora de LLMs.

Mediante *TestBuddy*, los desarrolladores podrán optimizar la creación de pruebas unitarias, pues la extensión generará automáticamente conjuntos de pruebas tan pronto como se finaliza un fragmento de código. Si el desarrollador desea ajustar o mejorar las pruebas generadas, *TestBuddy* también permitirá una interacción colaborativa, en donde el programador sugiere condiciones específicas para nuevas pruebas. Esta característica promueve una mayor sinergia entre el desarrollador y la extensión.

Además de aumentar la eficiencia y la rentabilidad, *TestBuddy* se enfocará en la automatización para ofrecer una solución integral. Facilitará la generación continua de pruebas unitarias, permitiendo que el equipo de desarrollo se concentre en tareas más complejas y creativas, mientras se asegura la calidad del código.

La aplicación de *TestBuddy* tendrá un impacto significativo en el desarrollo de software, ya que abordará la necesidad crítica de pruebas oportunas y precisas. Al promover una cultura de Desarrollo Basado en Pruebas (TDD), *TestBuddy* contribuirá a reducir la acumulación de deuda técnica y garantizará un producto final de mayor calidad.

En resumen, *TestBuddy* representa una propuesta innovadora y relevante que maximizará el potencial de los LLMs en la industria del desarrollo de software. Al automatizar la creación de pruebas unitarias y fomentar la colaboración con el desarrollador para ajustar las pruebas, *TestBuddy* mejorará la eficiencia, la calidad y la rentabilidad de los proyectos de software. Esta extensión será un valioso aliado que aumentará la productividad y acelerará el proceso de desarrollo al reducir el tiempo dedicado a las pruebas manuales.

**Adjunte un diagrama de infraestructura y flujo del proceso de la solución propuesta.**

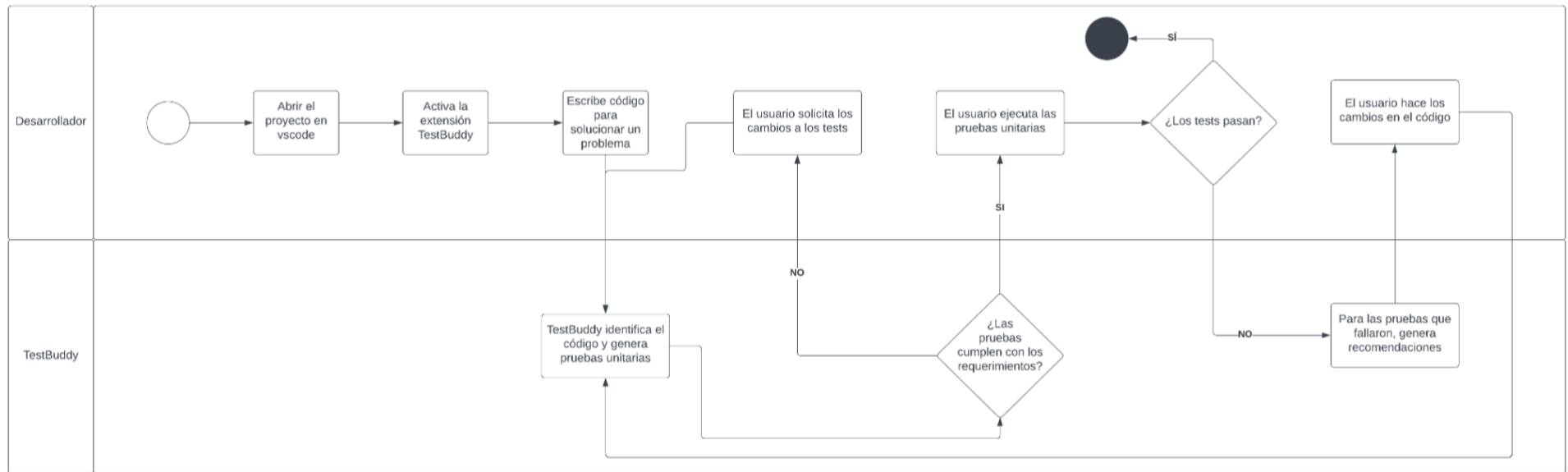


Figura 1: Diagrama de flujo de proceso de TestBuddy

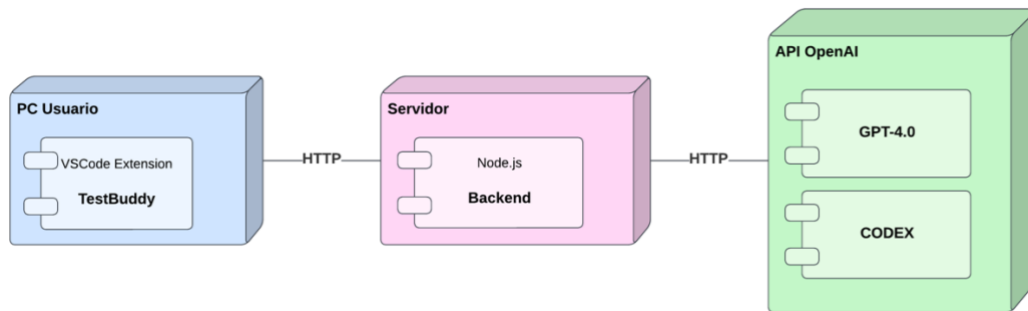


Figura 2: Diagrama de despliegue para la arquitectura de TestBuddy

- Describa las herramientas a usar y el por qué su elección.

**VSCode (Visual Studio Code):** La elección de VSCode como el entorno de desarrollo principal se justifica por su amplia popularidad y adopción en la comunidad de desarrolladores. Dado que *TestBuddy* busca ser una herramienta accesible y útil para una amplia audiencia de desarrolladores, es crucial utilizar un IDE ampliamente utilizado y con una gran cantidad de usuarios. Además, VSCode cuenta con una extensa gama de extensiones y soporte de la comunidad, lo que facilitará la integración y promoverá la escalabilidad del proyecto.

**JavaScript (JS):** El uso de JavaScript como lenguaje de programación principal se basa en la experiencia y habilidades del equipo de desarrollo. Al ser competentes en este lenguaje, el equipo puede desarrollar la extensión de manera más eficiente, permitiendo un desarrollo ágil y rápido. Además, JavaScript es una opción versátil y bien establecida que se integra bien con el entorno de desarrollo de VSCode y otras tecnologías.

**OpenAI:** La incorporación de OpenAI en la extensión TestBuddy proporciona el poder y las capacidades de inteligencia artificial necesarias para la generación automatizada de pruebas unitarias. OpenAI ofrece una amplia variedad de modelos de lenguaje avanzados, lo que permitirá al equipo acceder a tecnologías de vanguardia en inteligencia artificial y aprovechar su potencial para mejorar la eficiencia y precisión en la creación de pruebas.

**GPT (Generative Pre-trained Transformer):** La elección de GPT como uno de los modelos de lenguaje clave en TestBuddy es fundamental para garantizar la generación precisa y contextual de pruebas unitarias. GPT es una arquitectura probada y reconocida en el campo de procesamiento de lenguaje natural, y su capacidad para generar texto coherente y relevante será valiosa para el enfoque del proyecto.

**CODEX:** El uso de CODEX, otro modelo avanzado de OpenAI, en la extensión TestBuddy enriquecerá aún más sus capacidades de generación de pruebas unitarias. CODEX ha sido diseñado específicamente para comprender y producir código, lo que lo convierte en una herramienta poderosa para el desarrollo de software. Su integración en TestBuddy permitirá una generación de pruebas más especializada y adaptada al código fuente específico.

**Azure:** Se ha optado por utilizar la plataforma Azure por dos motivos fundamentales. En primer lugar, contamos con miembros del equipo que poseen experiencia previa en esta nube, lo que nos permitirá centrarnos plenamente en el desarrollo de la solución sin

distraernos en aspectos de infraestructura. En segundo lugar, las bases de la Hackathon sugieren específicamente el uso de esta plataforma, lo que nos proporciona una oportunidad idónea para aprovechar sus recursos y ventajas en el contexto de nuestro proyecto.

- Liste los integrantes del equipo, teniendo en cuenta: Nombre y apellidos, Identificación (cédula), Universidad, correos electrónicos, teléfono de contacto.

Nombre	Identificación	Universidad	Correo electrónico	Celular	LinkedIn
Juliana Prieto Arcila	1000285124	Universidad de los Andes	<a href="mailto:j.prieto@uniandes.edu.co">j.prieto@uniandes.edu.co</a>	3177007940	<a href="https://www.linkedin.com/in/julianaprietoa/">https://www.linkedin.com/in/julianaprietoa/</a>
Sofia Torres Ramírez	1000150341	Universidad de los Andes	<a href="mailto:s.torres21@uniandes.edu.co">s.torres21@uniandes.edu.co</a>	3222165957	<a href="https://www.linkedin.com/in/sofiatorresr/">https://www.linkedin.com/in/sofiatorresr/</a>
Paula Daza Díaz	1034396392	Universidad de los Andes	<a href="mailto:p.dazad@uniandes.edu.co">p.dazad@uniandes.edu.co</a>	3015522084	<a href="https://www.linkedin.com/in/paudazad">www.linkedin.com/in/paudazad</a>
Daniel Reyes Ramírez	1193428084	Universidad de los Andes	<a href="mailto:df.reyesr@uniandes.edu.co">df.reyesr@uniandes.edu.co</a>	3165363691	<a href="https://www.linkedin.com/in/danielkings42/">https://www.linkedin.com/in/danielkings42/</a>
Camilo Andres Salinas Martinez	1073253985	Universidad de los Andes	<a href="mailto:ca.salinas@uniandes.edu.co">ca.salinas@uniandes.edu.co</a>	3115904947	<a href="https://www.linkedin.com/in/rebrandtsx">https://www.linkedin.com/in/rebrandtsx</a>

- Adjunte cualquier documento adicional o enlace relevante que respalde su propuesta (opcional):

## Bibliografía

Delgado, D., & Martinez, A. (2023, January 16). *Cost Effectiveness of Unit Testing: A Case Study in a Financial Institution*. IEEE Xplore. Retrieved July 28, 2023, from <https://ieeexplore.ieee.org/abstract/document/6681377>

Devmate. (2023, 04 26). *The three keys to minimizing the cost of unit testing*. Devmate. Retrieved July 28, 2023, from <https://www.devmate.software/post/the-3-keys-to-minimizing-the-cost-of-unit-testing>

Lteif, G. (2023, February 25). *Unit Testing: Is It Worth the Effort?* Operational Excellence in Software Development Software Development. Retrieved July 28, 2023, from <https://softwaredominos.com/home/software-design-development-articles/unit-testing-is-it-worth-the-effort/>

Nedelcu, B. (2020, May 7). *The price of software: Unit testing*. DEV Community. Retrieved July 28, 2023, from <https://dev.to/bogdanned/the-price-of-software-unit-testing-59i9>

Roopa, M. (2019, 09 2). *Software Engineering based Cost Reduction Techniques for Different Testing Models*. IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/8822134>



Departamento de Ingeniería  
de Sistemas y Computación



DEVSavant



37 años  
1982-2019

Escuela de Ingeniería,  
Ciencia y Tecnología