|  |  |
| --- | --- |
|  | **Instituto Politécnico Nacional**  **Unidad Profesional Interdisciplinaria de Ingeniería campus Zacatecas** |
|  | **Área de ubicación para el desarrollo del trabajo**  Ingeniería en Sistemas Computacionales |
| **Línea de investigación**  Inteligencia Artificial |
| **Título del proyecto de Trabajo Terminal**  InnovaTRIZ: Plataforma Inteligente para la automatización de la innovación con TRIZ |
| **Presenta(n):**  Francisco Javier Calderón Corrales.  Arath Vite Rodríguez.  Raúl Eduardo Us Cardona. |
| **Director:**  Dra. Yesika Yuriri Rodríguez Martínez. |
|  | **Asesores:**  M.I.S. Julia Elena Hernández Ríos. |
|  | Zacatecas, Zacatecas a 02 de Septiembre de 2025 |

**Índices**

**Contenido**

[Descripción del proyecto. 1](#_Toc207863001)

[Objetivo general del proyecto. 2](#_Toc207863002)

[Objetivos particulares del proyecto. 2](#_Toc207863003)

[Marco metodológico. 2](#_Toc207863004)

[Cronograma de actividades. 9](#_Toc207863005)

[Bibliografía. 10](#_Toc207863006)

[Firmas. 11](#_Toc207863007)

[Autorización. 11](#_Toc207863008)

**Índice de tablas**

[Tabla 1 Comparación de metodologías 3](#_Toc207863010)

[Tabla 2 Comparación de ventajas y desventajas de en las metodologías agiles 5](#_Toc207863011)

**Índice de figuras**

[Figura 1 Metodología scrum 6](#_Toc207863017)

[Figura 2 Metodología Kanban 6](#_Toc207863018)

Descripción del proyecto.

InnovaTRIZ será una plataforma web que especialmente será diseñada para facilitar el aprendizaje y la aplicación práctica de la metodología TRIZ mediante la incorporación de inteligencia artificial. Esta herramienta busca ofrecer una solución que permita a estudiantes y profesores abordar la resolución de problemas técnicos de manera sistemática y efectiva. Los usuarios podrán describir los problemas técnicos que enfrentan, tras lo cual el sistema procederá automáticamente a realizar un análisis que se basa en la metodología TRIZ para identificar las contradicciones técnicas a dichos problemas o procesos. Posteriormente, se relacionarán estas contradicciones con los principios de la matriz de TRIZ, de tal manera que propone recomendaciones específicas y estructuradas que guíen al usuario hacia soluciones innovadoras.

La plataforma también facilitará la generación automática de soluciones mediante la consulta de una base de datos especializada y la aplicación de técnicas de inteligencia artificial. Estas soluciones serán presentadas de forma comprensible y adaptadas al contexto específico proporcionado por los usuarios. Además, los usuarios tendrán la oportunidad de evaluar y validar las soluciones propuestas, pudiendo solicitar alternativas adicionales según lo necesiten o bien lo prefieran.

También con InnovaTRIZ se podrán conocer casos de estudio relevantes de innovaciones exitosas alcanzadas mediante la metodología TRIZ, y diversas actividades interactivas de aprendizaje diseñadas para reforzar los conceptos de aprendizaje que sean adquiridos de la metodología TRIZ y promover su aplicación automatizada efectiva en situaciones prácticas.

El propósito fundamental de InnovaTRIZ es contribuir al sistema del proceso de la aplicación manual de la metodología TRIZ en diversas instituciones de ingeniería, proporcionando una herramienta accesible. Al mismo tiempo, InnovaTRIZ busca cerrar la brecha existente en la aplicación de la metodología TRIZ de forma tradicional, fomentando así una cultura y pensamiento de innovación y creativa en el ámbito educativo y profesional.

Objetivo general del proyecto.

Implementar una plataforma web que se basa en inteligencia artificial que logre apoyar el aprendizaje y la aplicación de la metodología TRIZ en la resolución de problemas de ingeniería, mediante un entorno interactivo que facilite la identificación de contradicciones y de recomendaciones precisas de principios de innovación.

Objetivos particulares del proyecto.

* Utilizar un modelo de procesamiento de lenguaje natural como herramienta para interpretar efectivamente los problemas descritos por los usuarios y facilitar así la sugerencia precisa de soluciones basadas en TRIZ.
* Implementar una base de datos con casos prácticos específicos que sirvan como apoyo para la correcta aplicación de los principios TRIZ en diferentes contextos.
* Desarrollar una interfaz web intuitiva y de fácil acceso que optimice la interacción de los usuarios con la plataforma, asegurando una experiencia fluida y efectiva.
* Integrar algoritmos de inteligencia artificial que permitan mejorar la precisión en la identificación de problemas técnicos y aumentar la efectividad de las soluciones recomendadas por InnovaTRIZ.

Marco metodológico.

El desarrollo del proyecto de Inteligencia Artificial basada en TRIZ se llevará a cabo bajo un enfoque ágil utilizando Scrum en combinación con Kanban. Esta elección metodológica permite un control flexible y visual de las tareas, con entregas incrementales que facilitan la retroalimentación continua y la adaptación a los cambios que se vayan presentando en el transcurso del trabajo.

Para fundamentar esta elección, es importante considerar cómo se posicionan estas metodologías frente a los enfoques tradicionales y orientados a objetos. Mientras que las metodologías tradicionales suelen ser rígidas y secuenciales, las orientadas a objetos introducen iteración y modelado visual, y las ágiles priorizan la adaptabilidad y el software funcionando. La siguiente tabla resume estas diferencias:

Tabla 1 Comparación de metodologías

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo | Metodologías | Características principales | Ventajas | Desventajas |
| Tradicionales | * Cascada * Modelo V | * Secuenciales y rígidas.  Mucha documentación. * Cambios costosos. * Adecuadas para proyectos con requisitos estables. * Enfoque lineal: análisis → diseño → implementación → pruebas → mantenimiento. | * Claridad en cada fase. * Buena trazabilidad y documentación. * Fácil de gestionar en proyectos pequeños o con requisitos bien definidos. | * Poco flexibles ante cambios. * Largas esperas antes de tener un producto funcional. * Riesgo alto si los requisitos cambian durante el desarrollo. |
| Orientadas a Objetos | * Booch  OMT (Object * Modeling Technique) * RUP (Rational Unified Process) | * Basadas en objetos y clases. * Uso de UML y modelado visual. * Desarrollo iterativo e incremental. * Promueven reutilización de componentes. * Énfasis en la arquitectura del sistema y patrones de diseño. | * Facilitan el mantenimiento y escalabilidad. * Buena integración con el paradigma de programación orientado a objetos. * Favorecen el análisis y diseño visual. | * Complejas de aplicar en proyectos pequeños. * Requieren mayor curva de aprendizaje. * Pueden generar sobrecarga de documentación y modelado. |
| Ágiles | * Scrum * Kanban * XP (Extreme Programming) * Crystal | * Iterativas e incrementales. * Flexibilidad ante cambios. * Priorizan software funcionando sobre documentación extensa. * Alta interacción con el cliente. * Retroalimentación continua. * Orientadas a equipos colaborativos y autoorganizados. | * Alta capacidad de adaptación. * Entregas rápidas y valor temprano al cliente. * Mejor comunicación dentro del equipo. * Reducción de riesgos gracias a las iteraciones cortas. | * Requieren compromiso y disciplina del equipo. * Documentación limitada (puede ser un problema en entornos académicos o regulados). * Difíciles de implementar si no hay comunicación constante con el cliente. |

Como puede observarse, el enfoque ágil resulta más conveniente para proyectos de investigación y desarrollo como *InnovaTRIZ*, donde se requiere flexibilidad, entregas rápidas y validación constante con usuarios.

El modelo de trabajo se organiza en fases iterativas que abarcan actividades de investigación, desarrollo de producto y administración del proyecto. En la fase inicial se realizarán la revisión documental, el análisis de los principios de TRIZ y la estructuración de la base de conocimientos que servirá como datos de aprendizaje para la Inteligencia Artificial. Posteriormente, en la fase de diseño conceptual y técnico, se definirán la arquitectura del sistema, las herramientas tecnológicas necesarias y los prototipos de la interfaz.

El desarrollo del modelo de IA contempla la implementación de módulos para la detección de contradicciones, el motor de razonamiento con base en los principios TRIZ y la integración de técnicas de procesamiento de lenguaje natural. En paralelo, se desarrollará la plataforma de software que servirá como interfaz de interacción con los usuarios, incorporando pruebas de funcionalidad y usabilidad de manera continua.

Dado que dentro de las metodologías ágiles existen varias opciones, se evaluaron algunas de las más relevantes, como Scrum, Espiral y Kanban, considerando sus características, ventajas y limitaciones. La siguiente tabla muestra una comparación entre ellas:

Tabla 2 Comparación de ventajas y desventajas de en las metodologías agiles

|  |  |  |  |
| --- | --- | --- | --- |
| Metodología | Características | Ventajas | Desventajas |
| Scrum | * Iterativa, basada en sprints (2–4 semanas). * Roles definidos (Scrum Master, Product Owner, Equipo). | * Entregas rápidas e incrementales.  Flexibilidad ante cambios. * Buena organización del equipo. | * Requiere disciplina en roles y reuniones. * Puede ser pesado para equipos pequeños. |
| Espiral | * Iterativa y basada en gestión de riesgos. Cada ciclo incluye: planificación → análisis de riesgos → desarrollo → evaluación. | * Fuerte análisis de riesgos. * Buena documentación. * Útil en proyectos complejos. | * Más lento y burocrático. * Menos ágil ante cambios rápidos. |
| Kanban | * Flujo de trabajo visual mediante tableros. No usa sprints ni roles estrictos. | * Simple de aplicar. * Visual y flexible. * Fácil de combinar con otras metodologías. | * No define tiempos de entrega. * Riesgo de desorganización si no se gestiona bien. |

A partir de este análisis, se concluyó que Scrum aporta la estructura de trabajo iterativa y organizada, mientras que Kanban añade la flexibilidad y la visualización del avance de tareas, lo cual resulta fundamental en un proyecto que combina investigación, desarrollo de software e integración de técnicas de Inteligencia Artificial.

Durante todo el proyecto se aplicará un esquema de administración ágil mediante tableros Kanban, en donde las tareas estarán clasificadas según su estado (backlog, stories, to do, in progress, to verify, done). Esto permitirá mantener un control del avance sin perder la flexibilidad propia de un proceso de investigación y desarrollo. Asimismo, se llevará un control de versiones mediante repositorios de código (GitHub), con la finalidad de garantizar la organización de archivos, la trazabilidad de cambios y la entrega de líneas base en puntos definidos del proyecto.

Figura 1 Metodología scrum

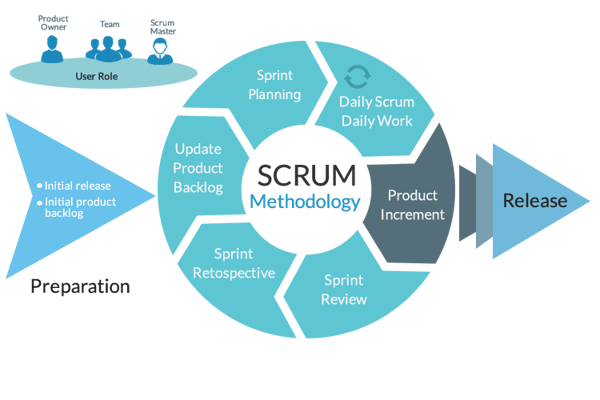
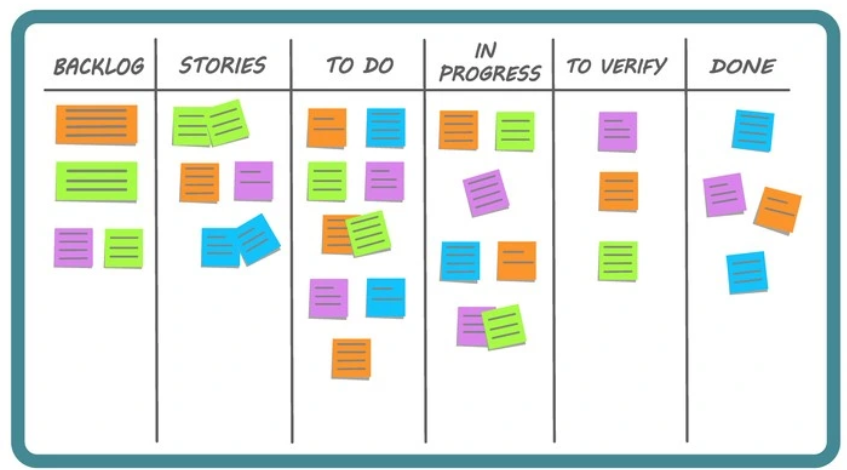


Figura 2 Metodología Kanban



En el marco de la metodología de desarrollo ágil Scrum, el proyecto se ha estructurado en cinco Sprints principales más un Sprint inicial de planificación. Cada Sprint cuenta con objetivos claros, actividades específicas y un entregable que contribuye al avance incremental del proyecto. A continuación, se describe y justifica cada uno de ellos:

* Sprint 0 – Planificación (28 agosto – 5 de septiembre):  
    
  Este sprint tiene como objetivo preparar el plan metodológico y el cronograma inicial del proyecto, validarlos con los asesores y directores, y finalmente obtener el dictamen de aceptación del anteproyecto. Se desarrolla entre el 28 de agosto y el 5 de septiembre de 2025. Durante este periodo se redacta la primera versión del plan metodológico, se construye un cronograma en formato Gantt que incluye los Sprint del 1 al 5 con sus dependencias y días no laborables, y se lleva a cabo una reunión con el cliente y el asesor para confirmar objetivos, riesgos y acuerdos. Posteriormente, los asesores revisan la documentación y se realizan los ajustes correspondientes para concluir con la entrega y archivo del dictamen de aceptación. Los entregables claves son el documento del plan metodológico versión 1.0, el cronograma detallado, la minuta de la reunión, las observaciones incorporadas y el dictamen firmado. El sprint se considera completado cuando toda la documentación está revisada, archivada y el repositorio actualizado. Entre los principales riesgos se encuentran la dificultad de coordinar agendas con el cliente y el asesor, observaciones tardías que retrasen la aceptación, y ambigüedad en el alcance que genere retrabajo. Como parte del marco Scrum, se realizan las ceremonias de planeación, reuniones diarias, revisión del sprint y retrospectiva.
* Sprint 1 – Levantamiento de requerimientos y prototipos (6 – 29 de septiembre):  
    
  El objetivo de este sprint es tener el sprint backlog priorizado y completamente desglosado. En esta etapa se preparan y publican las historias de usuario seleccionadas, las cuales se dividen en tareas con sus respectivas estimaciones, dependencias y responsables. El equipo se reúne de forma diaria para dar seguimiento al progreso, identificar bloqueos y planear el día. Al cierre del sprint, se organiza una reunión de revisión para presentar los resultados obtenidos y se lleva a cabo una retrospectiva para definir las mejoras del proceso. Como entregables se espera contar con el sprint backlog documentado, la lista de tareas asociadas a cada historia con estimaciones claras, el registro de bloqueos detectados y las minutas de las reuniones de revisión y retrospectiva. Para iniciar este sprint se requiere que las historias cumplan con la definición de listo, es decir, que tengan objetivos claros, criterios de aceptación preliminares, dependencias identificadas, referencias necesarias y riesgos anotados. Una historia se considera completada cuando está implementada y probada, el backlog está actualizado y los riesgos tienen acciones de mitigación aplicadas. Entre los principales riesgos están la ambigüedad en los requerimientos, la posibilidad de realizar estimaciones poco realistas y los bloqueos externos que retrasen el avance.
* Sprint 2 – Plan de riesgos y TRIZ inicial (1 – 6 de octubre):  
    
  Este sprint tiene como meta principal actualizar y priorizar el product backlog, además de formalizar el plan de riesgos. En esta fase el Product Owner, junto con los stakeholders, define el orden de prioridad de las historias y el equipo se encarga de elaborar la matriz de riesgos que incluye probabilidades, impactos y niveles de riesgo, así como las acciones de mitigación correspondientes. También se extrae el nuevo sprint backlog con las tareas que se ejecutarán en el siguiente ciclo. Durante este periodo se celebran reuniones con el cliente, el director o el asesor para obtener retroalimentación y se llevan a cabo las ceremonias de revisión y retrospectiva. Como entregables se generan un product backlog actualizado y priorizado, una matriz de riesgos documentada, un tablero de trabajo con las tareas del sprint y las minutas de todas las reuniones realizadas. El sprint se considera finalizado cuando el backlog está completo y priorizado, la matriz de riesgos se encuentra publicada, los registros de bloqueos en dailies están documentados y las acciones de mejora están definidas y archivadas. Los riesgos de esta etapa incluyen la falta de identificación adecuada de riesgos, la desalineación entre el equipo y el cliente en cuanto a prioridades y la ausencia de métricas claras para la priorización de historias.
* Sprint 3 – Diseño del sistema + IA basica (9 de octubre – 21 de noviembre):  
    
  Durante este Sprint se lleva a cabo el diseño de la solución. Entre las actividades destacan la definición de la arquitectura del sistema, el diseño de la base de datos, la elaboración de diagramas UML (clases, componentes, casos de uso, secuencia, entre otros) y el diseño de prototipos de la aplicación. Al concluir, se realiza una reunión de revisión con el cliente y el asesor. La justificación de este Sprint es que constituye la fase en la cual los requerimientos previamente definidos se transforman en una propuesta técnica concreta, que servirá de guía para la futura implementación.
* Sprint 4 – Pruebas y versión beta (22 de noviembre – 5 de diciembre):  
    
  Este Sprint se orienta a garantizar la calidad del sistema desarrollado. Para ello, se elabora el plan de pruebas, se ejecutan pruebas con la plantilla correspondiente, se registran los resultados en el control de pruebas y se construye la matriz de trazabilidad para asegurar la cobertura de los requerimientos. Asimismo, se realiza una reunión de validación con el cliente y el asesor. La justificación de este Sprint se fundamenta en la importancia de verificar y validar que el sistema cumple con los requerimientos establecidos, asegurando su funcionalidad y confiabilidad.
* Sprint 5 – Cierre y entrega final (diciembre – enero 2026):  
    
  El último Sprint está destinado al cierre del proyecto. Incluye la entrega del reporte final a los asesores y a la maestra de Trabajo Terminal, así como la preparación y exposición de la presentación final ante el jurado correspondiente. La justificación de este Sprint radica en que representa la consolidación de los entregables obtenidos en los Sprints anteriores y su formalización mediante la entrega documental y la defensa académica del trabajo realizado.

Cronograma de actividades.

El cronograma de actividades debe contener la calendarización de las actividades de investigación, desarrollo de productos y administración del proyecto de Trabajo Terminal. Para cada actividad del cronograma debe de especificarse la persona o personas responsables de realizar la actividad, la calendarización planeada y el tiempo real empleado hasta la conclusión de la actividad, así como la indicación del porcentaje de avance.

Se sugiere presentar el cronograma con una gráfica de Gantt, y será utilizado para la administración de las acciones realizadas o por realizar durante el desarrollo del proyecto de Trabajo Terminal, así mismo deberán incluir las actividades relacionadas con la estrategia de control de versiones (definición de la organización o configuración de los archivos y carpetas, y establecer fechas de entregas de líneas base).

Bibliografía.

Deberá incluir las fuentes de consulta utilizadas para el desarrollo del proyecto de Trabajo Terminal, las cuales son referenciadas en el contenido del protocolo. Se sugiere utilizar cualquiera de los formatos: APA, ACM, Chicago o IEEE.

|  |  |
| --- | --- |
| [1] | O. L. Lodoño Palacio, L. F. Maldonado Granados y L. C. Calderón Villafánez, «Gu{ía para constuir Estados del Arte.,» *International Corporation of networks of Kmowledge,* p. 39, 2014. |
| [2] | R. Hernández Sampieri, C. Fernández-Collado y P. Baptista Lucio, Metodología de la Investigación, Ciudad de México: Mc. Graw Hill, 2006. |

Firmas.

En esta sección se mostrarán los nombres y las firmas de los alumnos responsables del desarrollo del proyecto de Trabajo Terminal.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Alumno 1. | Alumno 2. |
|  |  |

Autorización.

Por medio del presente autorizo la impresión y distribución del marco metodológico y cronograma de actividades, toda vez que lo he leído, comprendido en su totalidad, y estar de acuerdo con su desarrollo.

Atentamente;

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Nombre y firma del director del proyecto de TT | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Nombre y firma del asesor 1. | Nombre y firma del asesor 2. |