

# Globant ➤

## SENA Soft 2025 Synthetic Edition

Construyendo el futuro del trabajo:  
humanos y máquinas inteligentes



## LINEAMIENTOS DESARROLLO INTEGRAL DE SOFTWARE (MVP)

### A. INTRODUCCIÓN

Estamos presenciando una revolución en el desarrollo de software desde que se publicó el paper «*Attention Is All You Need*» que introdujo la arquitectura del **Transformer** ([Vaswani et al., 2017](#)).

La IA llegó y cambió las reglas: hoy cualquiera puede escribir, comprender y validar código con ayuda de un LLM. Las barreras de entrada cayeron y usar un lenguaje o framework ya no es una limitante sino que es una elección estratégica.

Esta nueva etapa viene acompañada de una transformación en la propia naturaleza del software. Se trata de una «*Nueva Abstracción*»: no sólo un nivel más alto de generalización y modelado, sino que las soluciones se han convertido en no-deterministas por el uso de los modelos generativos ([Fowler, 2025](#)).

Este año, SENASoft es una oportunidad única para explorar el *Vibe Coding* ([Karpathy, 2025](#)) —esa nueva forma de crear prototipos funcionales con agilidad, colaboración e IA como copiloto— y volver así a lo esencial: resolver problemas reales, crear valor y pensar con claridad. No basta con saber programar. Se necesita criterio, adaptabilidad y visión integral.

Buscamos aprendices que hayan aceptado el reto de enfrentar el ciclo completo del desarrollo de software, colaborando en equipo con propósito, criterio y apoyo tecnológico. Para lograrlo, podrán apoyarse en herramientas de IA, aplicando técnicas como el *Prompt Engineering* ([Wei et al., 2022](#)) y también enfoques más recientes como el *Context Engineering* ([Karpathy, 2025](#)), con el objetivo de construir soluciones que las personas amen usar ❤️

Esta categoría reta a los equipos a recorrer el ciclo del desarrollo de software: analizar, diseñar, codificar, probar y entregar un MVP funcional. No importa si es web, móvil o de escritorio; ni el lenguaje, ni el framework, ni si es de uso libre o propietario. Lo más importante es crear software que funcione, resuelva un problema real y use ingeniosamente las capacidades tecnológicas actuales. No se trata sólo de ideas: buscamos MVPs con usabilidad inteligente, valor tangible y ejecución clara.

Este documento define los lineamientos técnicos de la categoría Desarrollo Integral tanto para instructores como para competidores. A partir de ello se diseña la ruta de habilitación y talleres, para fortalecer los conocimientos de los equipos y guiarlos en la resolución del reto.

### B. OBJETIVO

Establecer los requisitos técnicos y de conocimiento que guiarán las pruebas de la categoría Desarrollo Integral en SENASoft 2025, definiendo las competencias a desarrollar, en línea con la formación por competencias laborales impartida por el SENA y habilitadas especialmente para esta hackathon.

### C. TEMÁTICA

Este año vamos a reflexionar sobre los agentes de IA que llegaron para quedarse. No piensan en el sentido humano de la palabra, pero lo imitan muy bien. Cada equipo en la hackathon va a trabajar con ellos, sin delegar lo esencial: *personas y máquinas que imitan la inteligencia tomando decisiones juntos*. Resolverán un reto de negocio con habilidades técnicas y colaboración efectiva entre humanos e IAs.

#### D. DEL PROBLEMA AL PROTOTIPO: UN RECORRIDO GUIADO

Este año tendremos una hackathon contada por capítulos. Desde la comunicación del kickoff a toda la comunidad de aprendices invitada hasta la premiación al mejor MVP construido y presentado.



Fase	Capítulo	Descripción	Audiencia
<b>El despertar del equipo agéntico</b>	1. Kickoff 2. Lakes of reflection 3. Portal of courage 4. Crystal tower of selection 5. Circle of teams	1. Transmisión en vivo para todo Colombia explicando el reto. 2. Uso de agentes IA ( <b>Quillano</b> y <b>Magnus</b> ) previo a inscribirse. 3. Registro de aprendices (sin límite x centro) y ruta habilitadora. 4. Evaluación y selección de quienes irán a la hackathon. 5. Creación de equipos basados en el perfilamiento de <b>Magnus</b> .	Aprendices
<b>La Batalla del MVP</b>	6. Garden of connection 7. AI Arena 8. Hall of the Makers 9. The pier to Beyond	6. Socialización entre equipos inter-centro de todo Colombia. 7. Encuentro presencial de todos los participantes para el MVP. 8. Presentación de la solución a los jurados de Globant. 9. Premiación de equipos ganadores y próximos pasos.	Aprendices
<b>El Concilio de los Sabios</b>	10. The synthetic lab 11. The MCP alignment deck	10. Taller de un dataset sintético para cada equipo (presencial). 11. Charla ética sobre el uso de IA (Manifesto) y entrega de MCP.	Instructores /Aprendices

Inscríbete aquí para participar:  
[tinyurl.com/senasoft2025-inscribirme](https://tinyurl.com/senasoft2025-inscribirme)

Queremos conocer tu opinión:  
[tinyurl.com/senasoft2025-feedback](https://tinyurl.com/senasoft2025-feedback)

## E. REQUISITOS DEL STACK DE LA SOLUCIÓN

Cada equipo podrá elegir libremente su stack de desarrollo, siempre que sea funcional y consistente. Se acepta el uso de sistemas operativos como Windows, macOS o Linux, y cualquier servidor local de desarrollo como XAMPP, WAMP, Laragon, AppServ o IIS, según las necesidades del proyecto. Cada centro de formación deberá proveer tanto la máquina como las licencias necesarias para que sus aprendices participen en caso de ser seleccionados.

El reto consiste en idear, diseñar, desarrollar, probar y presentar al jurado un MVP funcional (web, móvil o de escritorio). ***El foco es el problema a resolver, no las herramientas para hacerlo:***

**E.1. Cualquier lenguaje de programación** del lado del servidor, como por ejemplo PHP (Laravel), Python (Django, FastAPI), Java (Spring Boot), Kotlin (Ktor, Spring), Ruby (Rails), Go (Gin), Rust (Rocket, Actix), C# (.NET) o JavaScript/TypeScript con Node.js (Express, NestJS).

**E.2. Cualquier IDE** para editar código y gestionar el desarrollo, con herramientas como Visual Studio Code, IntelliJ IDEA (Community), Sublime Text, Atom, NetBeans, Eclipse, Android Studio o Visual Studio Community, de acuerdo con su preferencia y el lenguaje elegido.

**E.3. Cualquier herramienta de diseño de interfaces** como Figma, Adobe XD, Sketch o Penpot, útil para crear prototipos visuales, estructurar la navegación, definir flujos de usuario y componentes. Es válido iniciar los diseños en papel o tablero, siempre que se migren a una herramienta digital.

**E.4. Cualquier framework de UI** como por ejemplo React, Angular, Vue.js, Svelte o similares para el lado del cliente. Se espera la creación de código de calidad con estándares web (HTML5, CSS3, JavaScript), y el uso de librerías o frameworks de estilos como Material Design, Bootstrap, Tailwind, SaSS o equivalentes, que ayuden a lograr una interfaz limpia, accesible y coherente.

**E.5. Cualquier motor de base de datos** puede incluirse en la solución, ya sea *relacional* (como MySQL, SQL Server, PostgreSQL, Oracle o MariaDB), *no relacional* (como MongoDB, Cassandra, DynamoDB o Redshift), *vectorial* (como Pinecone, Weaviate, Milvus o Qdrant), de *grafos* (como Neo4j, ArangoDB o Amazon Neptune), o basado en tecnología *blockchain* (como Ethereum, Hyperledger o IPFS).

**E.6. Cualquier herramienta de ejecución de pruebas** como Postman (para pruebas funcionales de APIs), JUnit (para pruebas unitarias en entornos Java) o Selenium (para pruebas automatizadas web).

**E.7. Cualquier plataforma web de control de versiones** como GitHub, GitLab o Bitbucket es *OBLIGATORIO*. Además de gestionar el código fuente, estas herramientas mostrarán el trabajo colaborativo (se requerirá lista de usuarios), los cambios realizados, la gestión de ramas y la trazabilidad del proyecto. El repositorio debe estar cargado en alguna de estas plataformas para su revisión. El almacenamiento exclusivo en local será penalizado, salvo en casos justificados de fuerza mayor.

**E.8. Cualquier herramienta de gestión del proyecto** con organización visual como Trello, Kanboard, Wekan, Taiga, OpenProject o SpreadSheet. Su uso facilita la asignación de tareas, el seguimiento de pendientes y la visibilidad del progreso del equipo. Se valorará positivamente cuando estas herramientas evidencian una división clara del trabajo y una ejecución ordenada del proceso.

**E.9. Cualquier entorno en la nube o contenedores** puede usarse para desplegar la solución, siempre que sea accesible para evaluación. Se recomienda usar Vercel, Render, Firebase, AWS, Azure, Google Cloud u otros, o entornos con Docker/Kubernetes. Las soluciones ejecutadas sólo en local serán penalizadas.

#### F. REQUISITOS DEL USO, INTEGRACIÓN Y DISEÑO DE IA

Cada equipo podrá integrar inteligencia artificial en su solución de forma libre, siempre que lo haga de manera coherente, funcional y alineada con el objetivo del reto. La IA no es un accesorio, es parte del corazón de la solución.

Se generará código con herramientas de IA y los Globers podrán verificar que los participantes entiendan su estructura, lógica, y puedan explicarla en cualquier momento que se les requiera.

El uso de inteligencia artificial se evaluará tanto en su integración técnica como en su pertinencia. A continuación, se enumeran las categorías de herramientas y enfoques permitidos (y en algunos casos requeridos) para crear, consumir y operar soluciones con IA durante la hackathon:

*F.1. Se incentiva el uso de técnicas de prompt y context engineering* para enriquecer la interacción con modelos de lenguaje. Estas prácticas pueden aplicarse directamente en el código, a través de flujos de automatización, o en la configuración de agentes conversacionales.

*F.2. Se incentiva el uso de asistencias de desarrollo basados en IA* incluyendo IDE-agente como Cline y Continue, así como herramientas ampliamente utilizadas como Kiro, Windsurf, Cursor, GitHub Copilot y Claude Code. Estas plataformas automatizan la generación, corrección y explicación de código directamente en editores como VS Code. Por ejemplo, Cline aprovecha el modelo Claude 3.7 Sonnet para planificar, editar archivos, ejecutar comandos y conectar con flujos externos, mientras que Claude Code se integra con terminales e IDEs para sugerencias y diffs inteligentes. El equipo debe mantener trazabilidad de su uso y comentar las secciones de código generadas con estas herramientas.

*F.3. Se incentiva el uso de herramientas visuales potenciadas por IA* para acelerar la creación de productos, como Loveable, Framer AI, Builder.io o similares, siempre que sirvan como base para refinar progresivamente el diseño y funcionamiento del MVP.

*F.4. Se requiere la integración de la solución con dos funcionalidades de IA:* una será con un MCP (*Model Context Protocol*) genérico y la otra será un agente para enlazar vía A2A (*Agent2Agent Protocol*), cuya funcionalidad deberá estar dentro del flujo de interacción con el usuario final. La correcta incorporación de estos elementos será evaluada. Los detalles técnicos de cada uno se darán antes de la hackathon.

*F.5. Se requiere que el diseño de la solución tenga capacidades agénticas (Feng et al., 2025)*, no simplemente de asistente. Esto implica que cada equipo debe integrar, al menos una vez, el consumo de un modelo de lenguaje (LLM) usando una API key, y activar dicho consumo como parte de un flujo automatizado mediante herramientas como n8n, Make (Integromat), Zapier, o integración propia equivalente. Se evaluará la forma en que este agente toma decisiones, responde con base en contexto, y completa tareas sin intervención humana directa.

## **G. RUTA HABILITADORA**

Los temas aquí listados serán abordados en la ruta habilitadora que preparará a los aprendices para el día de la competición. Se citarán a todos los aspirantes, no solo a los seleccionados, con el fin de que el conocimiento se extienda a todos los rincones de Colombia. No obstante, la asistencia y participación a dicha ruta será tenida en cuenta como una variable de selección a los competidores finales.

### **G.1. Tres-Amigos Sintético**

*Trabajo en equipo, diseño colaborativo, roles, ritmo y responsabilidades en un equipo híbrido con AI.*

Una charla introductoria que explora cómo deben organizarse los equipos durante la hackathon: todos colaboran, pero se requieren roles claros (DEV, BA, QC). Se explican los equipos sintéticos con analogías.

### **G.2. Pensar antes de codificar: diseño de producto y niveles de autonomía con IA**

*Del problema a la solución usando cartas de ideación agéntica, ¿solo responde o también actúa?*

Dinámicas virtuales y cartas físicas para ayudar a los equipos a idear una solución con comportamiento de agente. Explica también las diferencias entre asistentes, copilotos, agentes, guardrails, cómo sabemos que hay un human-in-the-loop que todavía está en control.

### **G.3. Datos sintéticos: calidad de software y cómo modelar lo que no existe**

*Generación y uso de datos para pruebas realistas, trazabilidad y responsabilidad cuando el código no es solo tuyo.*

Capacitación para comprender cómo y por qué crear datasets sintéticos personalizados para cada reto, probando donde parte del código es generado por IA. Se enfatiza el uso responsable y probado.

### **G.4. LLMs y Prompt Engineering en acción**

*Cómo hablarle bien a un modelo de lenguaje: GPT, Gemini, Mistral y más... ¿cuál es para qué?*

Recorrido por los modelos de lenguaje más populares, con ejemplos de uso en programación, análisis de producto y otras asistencias. Técnicas de escritura de prompts efectivos, con ejercicios prácticos.

### **G.5. Del boceto al código: interfaces creadas con ayuda de IA**

*Cómo generar pantallas útiles sin empezar desde cero*

Taller para usar herramientas como Loveable, Framer AI o Galileo para generar prototipos funcionales a partir de texto o bocetos. Ideal para avanzar más rápido.

### **G.6. IDEs con cerebro: el futuro ya llegó**

*Windsurf, Kiro, Cursor, Continue y otros copilotos inteligentes*

Demostración y guía de uso de IDEs potenciados por IA que escriben, editan, prueban y despliegan. Una sesión clave para liberar velocidad de ejecución.

### **G.7. MCP es el nuevo REST: cómo crear y consumir capacidades con flujos automáticos.**

*Tu solución no está sola: se espera que hable con otros servicios y combinarlo con flujos en n8n y Make.*

Sesión práctica sobre cómo construir y consumir un servidor MCP (Model Context Protocol) en cualquier stack. Se entrega un ejemplo real conectando APIs, acciones y agentes.

### **G.8. MVP en producción: lo mínimo para publicar en la nube**

*Cómo publicar tu MVP y base de datos sin complicarte la vida*

Una guía práctica para que los equipos puedan subir su aplicación y base de datos a la nube o plataformas como Vercel, Render, Railway o Firebase, incluso sin experiencia previa.

### **G.9. Hablar en público y presentar como un emprendedor**

*Cómo contar la historia de tu agente en minutos*

Prepara a los equipos para mostrar su solución con claridad, impacto y confianza frente al jurado.

Inscríbete aquí para participar:

[tinyurl.com/senasoft2025-inscribirme](http://tinyurl.com/senasoft2025-inscribirme)

Queremos conocer tu opinión:

[tinyurl.com/senasoft2025-feedback](http://tinyurl.com/senasoft2025-feedback)

## H. PROGRAMAS DE FORMACIÓN APLICABLES

Aprendices de los siguientes programas se pueden inscribir siempre que al momento de la competencia cumplan con los requisitos de los **lineamientos generales que el SENA comunica** aparte de este documento (trimestres válidos, edad, títulos, otros) y pertenecer a uno de estos programas:

- Tecnólogo: Análisis Y Desarrollo De Sistemas De Información
- Tecnólogo: Análisis Y Desarrollo De Software
- Técnico: Programación de Aplicaciones y Servicios Para La Nube
- Técnico: Programación de Software

## I. CRONOGRAMA DE LA HACKATHON

Momento	Competencia relacionada	Ejemplos de avances estimados
<b>Dia 1</b> <b>Análisis y Diseño de la solución agéntica</b>  <b>45%</b>	<ol style="list-style-type: none"> <li>1. Organizar el equipo con roles diferenciados y visión de equipo sintético.</li> <li>2. Analizar el problema de forma integral, considerando el usuario, el contexto y oportunidades de valor.</li> <li>3. Definir el comportamiento esperado del agente, su autonomía y la interacción con MCP y A2A.</li> <li>4. Diseñar la solución con claridad técnica y visión agéntica (arquitectura, flujos, IA, herramientas requeridas).</li> <li>5. Modelar adecuadamente los datos de entrada y salida.</li> <li>6. Diseñar la experiencia del usuario final y cómo interactúa con el agente.</li> <li>7. Preparar los recursos técnicos base para trabajar colaborativamente.</li> </ol>	<ul style="list-style-type: none"> <li>- Elección de roles y responsabilidades del equipo (BA, QA y Dev) con ayuda de las conversaciones previas con <b>Magnus</b>.</li> <li>- Usar las conversaciones previas con <b>Quiliano</b> para definir el prototipo a construir.</li> <li>- Definición del objetivo y el alcance del prototipo.</li> <li>- Diseño de los casos de negocio del roadmap.</li> <li>- Diseño de arquitectura, definición de la metodología de trabajo, definición del lenguaje(s) de programación propuesto(s) con ayuda de <b>Ada</b>.</li> <li>- Diseño de bases de datos.</li> <li>- Diseño de mockups, interfaz de usuario, experiencia de usuario con ayuda de <b>Ada</b>.</li> <li>- Creación del repositorio, revisión de requisitos técnicos del proyecto.</li> </ul>
<b>Dia 2</b> <b>Desarrollo del prototipo</b>  <b>25%</b>	<ol style="list-style-type: none"> <li>1. Implementar el prototipo de acuerdo con el diseño propuesto y el enfoque agéntico.</li> <li>2. Integrar correctamente los <b>componentes obligatorios</b>: MCP y agente A2A.</li> <li>3. Integrar al menos un LLM como <b>requisito obligatorio</b> en un punto clave usando apikay.</li> <li>4. Automatizar <b>al menos un</b> proceso mediante flujos con n8n, Make o similares.</li> <li>5. Ejecutar pruebas básicas para asegurar el funcionamiento de la solución.</li> </ol>	<ul style="list-style-type: none"> <li>- Desarrollo funcional de la solución al reto planteado, alineado con el comportamiento esperado del agente.</li> <li>- Consumo exitoso de MCP y canalización del A2A dentro del flujo del agente con ayuda de <b>Ada</b>.</li> <li>- Integración de un LLM dentro del flujo de automatización con ayuda de <b>Ada</b>.</li> <li>- Implementación de flujos que conectan componentes IA, interfaces, datos y decisiones con ayuda de <b>Ada</b>.</li> <li>- Documentación de pruebas unitarias y funcionales mínimas sobre el flujo principal.</li> </ul>
<b>Dia 3</b> <b>Entrega del prototipo</b>  <b>30%</b>  10% Sustentación 10% Producto final 10% Respuestas	<ol style="list-style-type: none"> <li>1. Entrega final del prototipo funcional y coherente con lo diseñado y desarrollado durante los días anteriores.</li> <li>2. Capacidad para responder con criterio a preguntas del jurado, evidenciando dominio del proyecto y trabajo en equipo.</li> <li>3. Interactuar en lengua inglesa de forma oral y escrita en tanta parte de la presentación final como le sea posible al equipo. Cuanto más la usen, más puntos tendrán.</li> </ol>	<ul style="list-style-type: none"> <li>- Afinamiento de los casos de negocio preparándose con ayuda de <b>Quiliano</b>.</li> <li>- Afinamiento de la solución propuesta.</li> <li>- Afinamiento y ejecución de las pruebas desarrolladas para la solución propuesta.</li> <li>- Elaboración de presentación de la solución propuesta preparándose con ayuda de <b>Ada</b>.</li> <li>- Presentación del prototipo a los jurados preparándose con ayuda de <b>Eleanor</b>.</li> <li>- Sustentación de la solución propuesta.</li> <li>- Respuesta a preguntas por parte del jurado.</li> </ul>

Inscríbete aquí para participar:

[tinyurl.com/senasoft2025-inscribirme](http://tinyurl.com/senasoft2025-inscribirme)

Queremos conocer tu opinión:

[tinyurl.com/senasoft2025-feedback](http://tinyurl.com/senasoft2025-feedback)

## J. CRITERIOS DE EVALUACIÓN

El proceso de evaluación para esta versión será liderado por los expertos de la empresa Globant combinado con puntajes dados por herramientas de IA preparadas para la ocasión. Estos son los ítems que se tendrán en cuenta en el proceso de evaluación:

### *J.1: Por el comité técnico (evaluación orgánica):*

1. Solución óptima al problema de acuerdo al problema planteado.
2. Documentación de los requerimientos funcionales.
3. Arquitectura planteada.
4. Diseño de UX/UI.
5. Modelamiento de datos (DER o MER).
6. Diagrama de clases (si aplica) y de secuencia.
7. Desarrollo en Backend y pruebas funcionales.
8. Desarrollo en Frontend y monolítico/microservicios.
9. Uso de autenticación JWT (Json Web Token).
10. Manejo de sistema de control de versiones.
11. Trabajo en equipo: evidenciado en herramienta de control de versiones.
12. Plan de pruebas y pruebas unitarias.
13. Despliegue de la solución en la nube.
14. Argumentación y presentación final de la solución frente al panel.

### *J.2: Por asistentes de IA (evaluación sintética):*

1. Puntaje dado al equipo por parte de **Quillano**.
2. Puntaje dado al equipo por parte de **Ada**.
3. Puntaje dado al equipo por parte de **Eleanor**.
4. Puntaje dado al equipo por parte de **Alan**:
  - a. Verificación de la integración con el MCP Tool.
  - b. Verificación de la integración con el agente via A2A.
  - c. Verificación de la integración con un LLM vía api key.
  - d. Verificación de consumo de tokens en USD (si aplica).

## K. CRITERIOS DE DESEMPEÑO

Para definir el equipo ganador, en caso de empate se dará un punto adicional, **en este orden**:

**K.1. Nube:** que la solución haya sido desplegada en una nube y tenga acceso público.

**K.2. Componentes:** que la solución contenga componentes de software desarrollados por el equipo, adicionales a los descritos en el documento de arquitectura y diseño.

**K.3. Funcionalidad:** que la solución contenga una funcionalidad extra, fuera de las descritas en el documento de especificación.

**K.4. Contenedores:** que la solución haya sido desplegada en alguna tecnología de contenedores.

**K.5. Documentación:** que exista un entregable de documentación adicional al comprometido en el documento de diseño, como por ejemplo la que podría generarse automáticamente a partir de los comentarios de código, pero también aplica para, Casos de Negocio (historias de usuario), Pruebas Funcionales, Pruebas no Funcionales, Metodología.

**K.6. Tiempo de entrega:** se verificará el tiempo de entrega en cada una de las jornadas asignando un punto en cada jornada a la pareja que haya entregado primero su prueba.

**K.7. App móvil:** desarrollo de algún tipo de solución móvil como complemento a la solución.