

**Laboratorio 8**  
**Exploración y Uso Avanzado de Plataformas IA, Repositorios**  
**Profesionales y Herramientas Globales para el desarrollo de IA**  
**y de SW**

**Materia:**

Profundización de inteligencia artificial

**Participantes:**

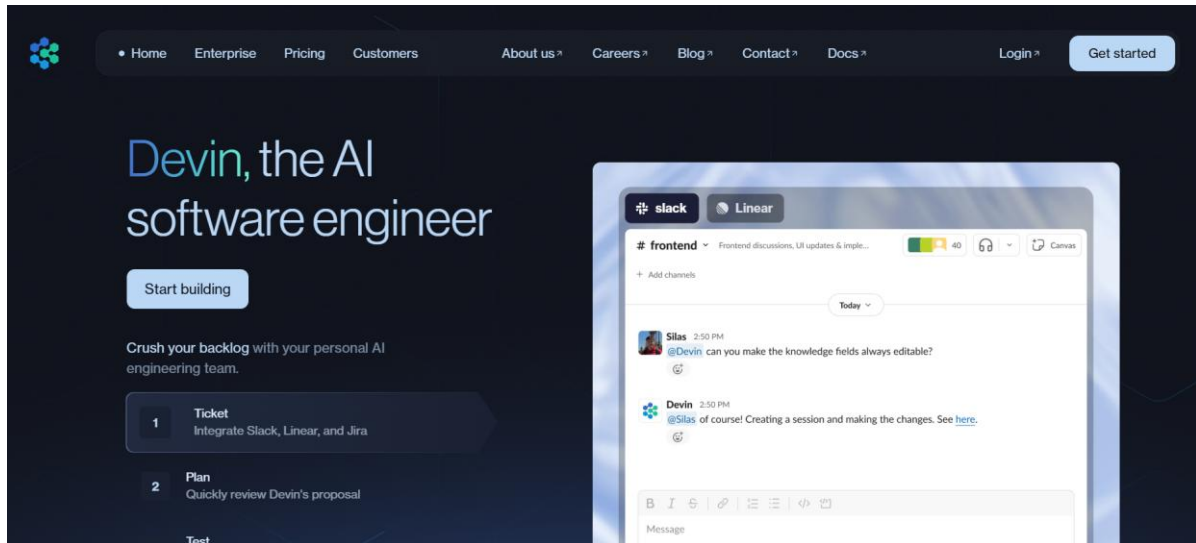
Ana Maria Navarro Fernandez

**Profesor:**

Carlos Betancourt correa

Universidad de Manizales  
ingeniería en sistemas y telecomunicaciones  
Manizales, Caldas, Colombia

## 2.14 Devin



## 1. Introducción

El laboratorio solicita analizar la herramienta Devin, presentada por Cognition Labs como un “AI Software Engineer”. A diferencia de los copilotos tradicionales (Copilot, Codeium, Claude Code), Devin se propone como un agente autónomo capaz de comprender tareas complejas de programación, planearlas, ejecutarlas, depurarlas y entregar resultados funcionales. Este informe cumple con los requisitos del laboratorio y desarrolla un análisis técnico y crítico sobre sus capacidades, limitaciones y aplicaciones educativas.

## 2. ¿Qué es Devin?

Devin es una inteligencia artificial que funciona como un ingeniero de software autónomo, es decir:

- Puede leer requerimientos completos.
- Puede crear un plan de trabajo (paso a paso).
- Puede escribir código en múltiples lenguajes.
- Puede abrir, editar y estructurar repositorios.
- Puede ejecutar pruebas, depurar errores y corregir fallas.
- Puede navegar en internet para buscar documentación.
- Puede terminar proyectos completos sin supervisión constante.

En otras palabras, Devin no solo sugiere código como Copilot: trabaja como un desarrollador que razona y actúa por sí mismo.

## 3. ¿Cómo funciona Devin? (según información pública)

Aunque su arquitectura no es abierta, se sabe que Devin combina:

## ✓ 1. Modelo de lenguaje avanzado

Probablemente una versión mejorada de modelos tipo GPT o Claude entrenados para:

- seguir instrucciones complejas,
- razonar en múltiples pasos,
- generar planes de acción,

- mantener contexto profundo.

### 2. Entorno virtual de trabajo (sandbox)

Devin puede usar:

- editor de código,
- terminal Linux real,
- navegador web interno,
- gestor de archivos,
- repositorios Git.

Este entorno permite que ejecute acciones reales, no solo texto.

### 3. Agente autónomo

Devin:

- analiza el objetivo,
- divide en subtareas,
- decide qué comando ejecutar,
- prueba su propio código,
- evalúa resultados,
- continúa trabajando.

Esto es un salto respecto a los copilotos actuales.

### 4. Métodos, técnicas y tipo de agente

- ✓ Razonamiento en múltiples pasos (multi-step reasoning)

Descompone tareas grandes en subtareas lógicas.

- ✓ Ejecución autónoma

No solo sugiere qué hacer: lo hace.

- ✓ Ciclo iterativo de mejora

Devin sigue un ciclo parecido a:

Entender el requerimiento

Planificar

Escribir código

Ejecutarlo

Analizar errores

Corregir

Repetir

- ✓ Agente de planificación + agente de ejecución

Parece usar una arquitectura de “dos cerebros”:

uno que razona,

uno que actúa.

### 5. Limitaciones actuales de Devin

1. No está disponible al público

Es la mayor limitación del laboratorio: solo se conoce por demostraciones.

2. Riesgo de errores graves

Al ejecutar comandos reales, podría:

borrar archivos,  
romper repositorios,  
introducir bugs invisibles.

3. Falta de explainability

No explica siempre por qué eligió una estrategia.

4. Necesita mucha supervisión humana en proyectos reales

No es seguro entregarle código crítico sin revisión.

5. Limitaciones éticas y laborales

Automatización excesiva.

Riesgos de dependencia en estudiantes.

Posible desplazamiento laboral.

### 6. Aplicaciones en educación

- ✓ Generación de ejemplos complejos para clases

Devin podría crear:

proyectos completos,  
aplicaciones simples para enseñanza,  
tareas con soluciones automáticas.

- ✓ Tutor en programación

Podría corregir y reescribir código erróneo de estudiantes.

- ✓ Ayuda en proyectos de grado

Podría asistir en:

diseño de APIs,  
conexión de bases de datos,  
análisis de errores,  
documentación.

- ✓ Simulador de entornos de desarrollo

Como agente autónomo, Devin puede enseñar:

flujo real de Git,  
pruebas unitarias,  
resolución de bugs.

### 7. Comparación con Copilot, Codeium y Claude Code

Herramienta	Tipo	Qué puede hacer	Nivel de autonomía	Limitaciones
<b>Devin</b>	Ingeniero autónomo	Planifica, ejecuta, prueba, corrige, navega	<b>Muy alto</b>	No disponible, riesgo alto
<b>GitHub Copilot</b>	Copiloto	Sugiere código	Bajo	No ejecuta, no depura
<b>Codeium</b>	Copiloto gratuito	Sugerencias rápidas	Bajo	Menos contexto
<b>Claude Code</b>	Analista	Revisa, explica, reestructura	Medio	No ejecuta en entorno real

### Conclusión

### comparativa:

Devin es el único que actúa como agente autónomo completo, pero su uso aún no es accesible ni seguro en contextos educativos sin supervisión.

### 8. Conclusiones finales

Devin representa un nuevo paradigma: IA que escribe, prueba y corrige código por sí sola.

Su nivel de autonomía supera ampliamente a los copilotos actuales.

No puede usarse todavía en el laboratorio porque no está disponible.

Es ideal para análisis conceptual, como exige esta actividad.

Puede transformar procesos educativos relacionados con programación.

Sus riesgos éticos, pedagógicos y técnicos exigen un uso responsable.

A futuro podría ser una herramienta clave para enseñanza asistida por IA.