

Innovación en Desarrollo de Software: Integrando IA con Codex y Claude Code en VSCode

Ingeniería | Ingeniería de sistemas | Design Thinking

Descripción

Este plan de clase está diseñado para estudiantes de posgrado en Ingeniería de Sistemas con el propósito de explorar y analizar cómo los asistentes de inteligencia artificial, específicamente Codex y Claude Code, pueden integrarse efectivamente en el flujo real de trabajo del desarrollo de software. Los estudiantes aprenderán a diferenciar las funcionalidades, ventajas, limitaciones y casos de uso ideales de cada asistente dentro del entorno de VSCode, profundizando en su impacto práctico en proyectos reales.

Este aprendizaje es relevante porque la automatización y la inteligencia artificial están transformando la manera en que se desarrolla software, exigiendo profesionales capaces de aprovechar estas herramientas para optimizar productividad y calidad. Al conectar el contenido con experiencias reales, los estudiantes podrán aplicar estos conocimientos en su práctica profesional, tomando decisiones informadas sobre qué asistente utilizar según el contexto y la necesidad del proyecto.

Además, se empleará la metodología Design Thinking para fomentar un aprendizaje activo centrado en la empatía con el usuario y la generación creativa de soluciones, desarrollando competencias críticas y analíticas en el uso de tecnologías emergentes.

Objetivos de Aprendizaje

- Comparar las características y funcionalidades de Codex y Claude Code integrados en VSCode.
- Analizar las ventajas y limitaciones de cada asistente de IA en el contexto del desarrollo de software.
- Demostrar cómo integrar efectivamente asistentes de IA en flujos reales de trabajo software.
- Aplicar la metodología Design Thinking para identificar casos de uso donde cada asistente aporta mayor valor.
- Evaluar críticamente la contribución de asistentes de IA en la mejora de procesos de desarrollo de software.

Recursos Necesarios

- Computadoras con VSCode instalado y acceso a internet
- Extensiones activas de Codex y Claude Code en VSCode (una por equipo)
- Proyector y pantalla para presentaciones
- Guía impresa con resumen comparativo de Codex y Claude Code
- Documento digital con casos reales de uso para análisis
- Pizarra blanca o digital y marcadores

- Herramienta colaborativa online (ej. Miro, Jamboard) para prototipado

Requisitos Previos

- Conocimientos previos en desarrollo de software y manejo básico de VSCode
- Familiaridad con conceptos básicos de inteligencia artificial y aprendizaje automático
- Experiencia previa en trabajo colaborativo y análisis crítico de herramientas tecnológicas
- Comprensión básica de metodologías ágiles y Design Thinking

Actividades

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión

Docente: Explica que la sesión busca entender cómo asistentes de IA como Codex y Claude Code pueden transformar el desarrollo de software, potenciando la eficiencia y calidad del trabajo diario. Destaca la importancia de conocer sus diferencias y casos de uso para una integración efectiva.

Estudiantes: Escuchan y preparan su mente para un análisis crítico y práctico.

Activación de conocimientos previos

Docente: Plantea la siguiente pregunta detonadora para debate breve: “¿Cuál ha sido su experiencia previa al usar asistentes de código o autocompletado? ¿Qué beneficios y problemas han encontrado?”

Estudiantes: Reflexionan individualmente 2 minutos y luego comparten en plenaria experiencias concretas (3-4 voluntarios breves), para conectar con el nuevo contenido.

Motivación y enganche

Docente: Presenta un dato impactante: “Según estudios recientes, el uso de asistentes de IA puede reducir hasta un 30% el tiempo de codificación en proyectos complejos. Hoy compararemos dos de las herramientas más innovadoras para que ustedes decidan cuál se adapta mejor a su estilo y proyecto.”

Estudiantes: Se motivan al entender la relevancia y aplicabilidad inmediata del tema.

Contextualización

Docente: Relaciona el contenido con su futura práctica profesional: “Como futuros líderes de desarrollo, saber cuándo y cómo usar estas herramientas puede marcar la diferencia en la entrega y calidad del software.”

Estudiantes: Reconocen la conexión directa con su desempeño profesional y académico.

Fase de Desarrollo

Tiempo estimado: 40 minutos

Presentación del contenido

Docente: Introduce brevemente las características generales de Codex y Claude Code, utilizando un esquema visual en pantalla para destacar diferencias clave. Explica que el enfoque será en la aplicación práctica más que en la teoría técnica profunda, invitando al uso de Design Thinking para descubrir usos y limitaciones.

Actividad 1: Empatizar y Definir - Análisis comparativo en grupos

- **Objetivo:** Comparar funcionalidades y casos de uso de Codex y Claude Code.
- **Instrucciones:**
 - Divide a los estudiantes en 3 grupos de 4 personas.
 - Entrega a cada grupo la guía impresa con comparación resumida y ejemplos de casos reales.
 - Solicita que identifiquen las fortalezas, limitaciones y posibles escenarios ideales para cada asistente.
 - Pide que definan una problemática común en desarrollo donde aplicarían cada herramienta.
- **Organización:** Grupos de 4 estudiantes.
- **Producto:** Listado en papel o digital con fortalezas, limitaciones y casos de uso.
- **Tiempo:** 15 minutos
- **Rol docente:** Facilita y orienta con preguntas como “¿Qué tipo de código genera cada asistente?”, “¿Qué limitaciones podrían afectar su flujo?”, “¿En qué escenarios creen que uno es más ventajoso que otro?”

Actividad 2: Idear y Prototipar - Simulación de integración en flujo de trabajo

- **Objetivo:** Demostrar la integración práctica de cada asistente en VSCode y prototipar un flujo de trabajo efectivo.
- **Instrucciones:**
 - En parejas, los estudiantes alternan el uso de Codex y Claude Code para resolver un problema sencillo de codificación asignado por el docente (por ejemplo, implementar una función o corregir un bug).
 - Mientras codifican, anotan ventajas y dificultades encontradas.
 - Luego, usando la herramienta colaborativa online, diseñan un esquema de flujo de trabajo integrando el asistente que consideren más adecuado para cada fase del desarrollo.
- **Organización:** Parejas.
- **Producto:** Esquema visual del flujo de trabajo y notas de experiencia.
- **Tiempo:** 20 minutos
- **Rol docente:** Observa, pregunta “¿Cómo facilita o dificulta el asistente su tarea?”, “¿Qué mejoras proponen para integrar mejor el asistente?”, y apoya con feedback inmediato.

Actividad 3: Evaluar - Presentación y retroalimentación

- **Objetivo:** Evaluar críticamente la integración de los asistentes desde la experiencia práctica.

- **Instrucciones:**

- Cada pareja presenta en plenaria su esquema y conclusiones en 3 minutos.
- El resto de la clase y el docente realizan preguntas y aportan comentarios.

- **Organización:** Plenaria.

- **Producto:** Presentación oral y discusión.

- **Tiempo:** 5 minutos

- **Rol docente:** Facilita la retroalimentación con foco en fortalezas y áreas de mejora, relacionando con los objetivos de aprendizaje.

Diferenciación

Para estudiantes que terminan antes: Se les invita a explorar funciones avanzadas de Codex o Claude Code, investigando y documentando una funcionalidad extra para compartir.

Para estudiantes que requieren apoyo: Se ofrece guía personalizada en la simulación práctica, con ejemplos concretos y aclaración de dudas sobre la interfaz y comandos.

Transiciones

Se conectan las actividades recordando que el análisis y diseño (Actividad 1) fundamenta la simulación práctica (Actividad 2), que a su vez alimenta la reflexión crítica y presentación (Actividad 3), cerrando el ciclo de Design Thinking.

Fase de Cierre

Tiempo estimado: 10 minutos

Síntesis

Docente: Solicita a cada estudiante escribir en un “ticket de salida” digital tres ideas clave aprendidas sobre la integración de asistentes IA en desarrollo, y un desafío que visualizan para su aplicación.

Estudiantes: Escriben sus ideas y desafíos de forma individual en el documento compartido.

Reflexión metacognitiva

Docente: Propone las siguientes preguntas para discusión rápida y reflexión interna:

- ¿Cómo influyen las diferencias entre Codex y Claude Code en la selección de asistentes para proyectos específicos?
- ¿Qué aspectos de la integración de asistentes IA consideran más críticos para mejorar en su práctica profesional?
- ¿De qué forma la metodología Design Thinking facilitó su comprensión y análisis del tema?

Estudiantes: Responden brevemente en plenaria y anotan reflexiones personales.

Retroalimentación

Docente: Proporciona comentarios inmediatos valorando la profundidad de análisis, la calidad de las presentaciones y la capacidad crítica demostrada, destacando la importancia del pensamiento estratégico en la integración de IA.

Transferencia

Docente: Conecta el aprendizaje con futuros desafíos en gestión de proyectos y selección tecnológica, invitando a aplicar lo aprendido en próximas actividades y en su entorno laboral.

Tarea o reto

Docente: Propone como tarea investigar una tercera herramienta de asistente IA emergente y preparar un breve informe comparativo con Codex y Claude Code, identificando potenciales escenarios de integración.

Estudiantes: Planifican y organizan tiempos para cumplir la tarea.

Evaluación

Tipo de evaluación: La evaluación es formativa y sumativa.

- **Formativa:** Durante la fase de desarrollo, a través de la observación directa, preguntas guía y retroalimentación continua en las actividades grupales y parejas.
- **Sumativa:** En el cierre, mediante la presentación oral, el ticket de salida y el informe de tarea complementaria.

Criterios de evaluación:

- Precisión y profundidad en la comparación de Codex y Claude Code (objetivo: comparar características y funcionalidades).
- Capacidad para identificar ventajas, limitaciones y casos de uso relevantes (objetivo: analizar ventajas y limitaciones).
- Demostración práctica en la simulación de integración en VSCode (objetivo: demostrar integración efectiva).
- Aplicación de Design Thinking para idear flujos de trabajo (objetivo: aplicar metodología Design Thinking).
- Reflexión crítica y argumentación en presentaciones y síntesis (objetivo: evaluar críticamente contribución de asistentes).

Instrumentos sugeridos: Rúbricas para presentaciones y esquemas, lista de cotejo para participación y análisis, observación directa, autoevaluación y coevaluación entre pares.

Evidencias de aprendizaje: Listados comparativos, esquemas de flujo de trabajo, notas de experiencia en codificación, presentaciones orales, tickets de salida y el informe de investigación complementaria.

Enriquecimientos

Recomendaciones - Tic_ia

Recomendaciones para Integrar Tecnología e Inteligencia Artificial en el Plan de Clase

Fase de Inicio (10 minutos)

- **Herramienta:** [Mentimeter](#) (Encuestas y preguntas en vivo)

Implementación: El docente lanza la pregunta detonadora sobre experiencias previas con asistentes de código usando Mentimeter para que los estudiantes respondan anónimamente en tiempo real mediante sus dispositivos. Luego, se exhiben las respuestas para abrir el debate.

Contribución a objetivos: Facilita la activación de conocimientos previos de manera rápida y participativa, fomentando la reflexión crítica sobre beneficios y limitaciones de asistentes IA en desarrollo de software.

Nivel SAMR: **Aumento** (mejora la interacción y recopilación de opiniones sin alterar la tarea de debate).

- **Herramienta:** [Prezi](#) (Presentación dinámica)

Implementación: El docente presenta datos impactantes y contextualiza la relevancia del tema mediante una presentación visualmente atractiva y no lineal que enganche a los estudiantes.

Contribución a objetivos: Motiva y contextualiza la sesión, resaltando la aplicabilidad de asistentes IA en el desarrollo profesional, preparando mentalmente a los estudiantes para el análisis profundo posterior.

Nivel SAMR: **Sustitución** (reemplaza presentaciones tradicionales con un formato digital mejorado).

Fase de Desarrollo (40 minutos)

- **Herramienta:** [Visual Studio Code con extensiones Codex y Claude Code](#)

Implementación: Los estudiantes trabajan directamente en VSCode, utilizando las extensiones oficiales para Codex (GitHub Copilot) y Claude Code para experimentar en tiempo real cómo cada asistente genera código, completamientos y sugerencias en distintos escenarios.

Contribución a objetivos: Permite una comparación práctica y vivencial entre los asistentes IA en un entorno profesional estándar, facilitando la comprensión de sus diferencias, fortalezas y limitaciones en el flujo real de trabajo.

Nivel SAMR: **Modificación** (rediseña la tarea tradicional de codificación incluyendo apoyo activo y contextualizado de IA, transformando la experiencia de aprendizaje).

- **Herramienta:** [Miro](#) (Pizarra colaborativa digital)

Implementación: Durante la actividad de Design Thinking, los estudiantes usan Miro para mapear colaborativamente casos de uso, ventajas y limitaciones detectadas en Codex y Claude Code, organizando ideas visualmente.

Contribución a objetivos: Facilita la colaboración y la estructuración visual del análisis crítico, fomentando el pensamiento de diseño y la síntesis colectiva de información relevante para la toma de decisiones.

Nivel SAMR: **Modificación** (permite replantear y enriquecer la actividad colaborativa con herramientas digitales interactivas).

Fase de Cierre (10 minutos)

- **Herramienta:** [Padlet](#) (Muro colaborativo para reflexiones finales)

Implementación: Los estudiantes escriben breves reflexiones finales sobre qué asistente IA consideran más adecuado para sus futuros proyectos y por qué, compartiéndolas en un Padlet accesible para toda la clase.

Contribución a objetivos: Promueve la reflexión individual y el intercambio de conclusiones, evidenciando la comprensión crítica sobre la integración de IA en el desarrollo de software.

Nivel SAMR: **Aumento** (facilita la expresión y recopilación de opiniones sin cambiar la naturaleza de la reflexión final).

- **Herramienta:** [Kahoot!](#) (Evaluación formativa gamificada)

Implementación: Se realiza un breve cuestionario en Kahoot! para evaluar la comprensión de diferencias, ventajas y limitaciones entre Codex y Claude Code, reforzando conceptos clave de forma dinámica.

Contribución a objetivos: Permite medir de manera ágil y motivadora el nivel de asimilación de los contenidos tratados, incentivando la participación activa hasta el final.

Nivel SAMR: **Aumento** (mejora la evaluación tradicional con tecnología interactiva sin cambiar la tarea de evaluación).