

A continuación, se presenta el documento de registro del proceso de creación de la aplicación web “¿Qué comemos hoy?”, en respuesta a la consigna del Trabajo Integrador del Bloque 1 de Codeki.

Trabajo Integrador: Creación de una Aplicación Web con Asistencia de IA

Proyecto: ¿Qué comemos hoy?

Curso: Bloque 1. Herramientas de IA para la Innovación en la Transformación Digital

1. Introducción: Idea Inicial y Objetivo de la Aplicación

El presente trabajo documenta el proceso de ideación, desarrollo y despliegue de una aplicación web funcional, concebida y ejecutada con la asistencia de herramientas de Inteligencia Artificial Generativa. El objetivo del proyecto es cumplir con los requisitos del Trabajo Integrador, demostrando la aplicación práctica de metodologías de prompting para la co-creación tecnológica.

La idea inicial fue desarrollar una solución tecnológica simple, lúdica y de utilidad práctica para el usuario. Así nació el concepto de

"¿Qué comemos hoy?", una aplicación web diseñada para ayudar a los usuarios a organizar las comidas del hogar.

El objetivo principal de la aplicación es sugerir recetas variadas al usuario, considerando tres variables clave: los ingredientes disponibles (con foco en frutas y verduras de estación), la cantidad de comensales y las preferencias del usuario. La aplicación fue concebida para ser de uso abierto en Argentina. La URL pública de la aplicación funcional es:

<https://xurl.es/recetas>.

2. Herramientas y Tecnologías Utilizadas

Para la realización de este proyecto se empleó un ecosistema de herramientas de IA y plataformas de desarrollo, entre las que se destacan:

- **Inteligencia Artificial Generativa:**
 - **Gemini 2.5 Pro:** Utilizado como asistente principal para la generación de ideas, estructuración de contenido, redacción de textos, generación de código (HTML, CSS, JavaScript), análisis de funcionalidades y planificación de la escalabilidad del proyecto.
 - **GitHub Copilot:** Empleado como asistente de programación para comprender procesos técnicos específicos, como el despliegue de la aplicación en un dominio público.
 - **Mistral AI, Together.ai (Kimi K2):** Utilizados para la evaluación y mejora del código generado, ofreciendo una segunda opinión técnica y sugerencias de optimización.

- **Llama 4 (Meta):** Empleado para la creación de imágenes de platos para las recetas.
- **Plataformas y Software de Desarrollo:**
 - **GitHub:** Utilizado para el control de versiones y como plataforma de alojamiento del código fuente y despliegue de la aplicación a través de GitHub Pages.
 - **Firebase Firestore:** Plataforma de base de datos recomendada por la IA y adoptada para la futura escalabilidad del recetario.
 - **TextEdit (iOS):** Editor de texto utilizado para la gestión inicial del código y la documentación.
 - **TinyPNG:** Herramienta para la optimización y reducción del peso de las imágenes generadas.

3. Proceso de Desarrollo Asistido por IA

El proyecto se estructuró en fases, siguiendo el ciclo de vida de un desarrollo de software y utilizando la IA como un "copiloto de programación" en cada etapa.

3.1. Fase 1: Ideación y Planificación

El proceso comenzó con una consulta a la IA para explorar conceptos de aplicaciones web funcionales y simples de desarrollar. La IA respondió con preguntas clarificadoras sobre el público objetivo y la temática, lo cual ayudó a refinar la idea. Se le presentó a Gemini un prompt más específico:

"imagino un ayudante en la organización de las comidas del hogar que sugiera menú con recetas variadas de acuerdo a la las verduras y frutas de estación y la cantidad de comensales".

La IA devolvió tres propuestas conceptuales, de las cuales se seleccionó la segunda, titulada

"¿Qué Cocino Hoy?". Esta idea fue elegida por su enfoque simple y espontáneo: el usuario ingresa uno o dos ingredientes que tiene disponibles y el número de comensales, y la aplicación le devuelve tres recetas rápidas y fáciles. La simplicidad de desarrollo, basada en un sistema de etiquetado de recetas por ingrediente, fue un factor decisivo.

Posteriormente, se utilizaron prompts para que la IA estructurara el contenido, los textos y un boceto inicial de la aplicación.

3.2. Fase 2: Desarrollo y Generación de Código

En esta fase, se le solicitó a la IA la generación del código base del proyecto. En respuesta, la IA proporcionó una maqueta funcional inicial en HTML con estilos CSS integrados.

A partir de esta base, se inició un proceso iterativo de refinamiento mediante prompts sucesivos para añadir funcionalidades clave:

- Inclusión de fotos para hacer las recetas más atractivas.

- Implementación de un filtro funcional basado en los ingredientes ingresados por el usuario.
- Definición de una estructura de receta completa, priorizando la simplicidad visual y la experiencia de usuario (UX).

3.3. Fase 3: Mejoras Iterativas y Depuración

Con una versión básica funcional, el proceso se centró en mejorar y robustecer la aplicación.

- **Análisis de Funcionalidad (Temporizador de Cocción):** Se exploró la idea de añadir un widget con un temporizador de cocción interactivo. Se le pidió a Gemini un análisis de pros y contras. La IA presentó una evaluación detallada, destacando la mejora en la UX pero señalando una complejidad técnica significativa: la suspensión de la ejecución de JavaScript en navegadores móviles cuando la pantalla se bloquea o se cambia de app, lo que haría al temporizador poco fiable. Basado en este análisis técnico, se tomó la decisión estratégica de **posponer esta funcionalidad para una futura versión 2.0**.
- **Mejora del Filtro de Búsqueda:** Para obtener una revisión externa del código, se utilizó el modelo Kimi K2 en la plataforma chat.together.ai. La IA sugirió diez mejoras, de las cuales se priorizó la optimización del filtro de búsqueda. Se le proporcionó a Gemini el fragmento de código específico para una búsqueda más flexible, y este lo integró correctamente en la lógica de JavaScript existente.

```
recetas.filter(r => ingredientesBuscados.some(ing => r.ingredientesClave.some(k => k.includes(ing) || ing.includes(k))));
```

- **Implementación de Escalado de Ingredientes:** Se decidió añadir una de las funcionalidades más potentes para una app de recetas. Mediante el prompt

"avancemos con el escalado de ingredientes", Gemini modificó el código para implementar esta lógica. La solución implicó añadir un campo `porcionesBase` a cada receta y crear una función en JavaScript que calcula un "factor de escala" para ajustar dinámicamente la cantidad de cada ingrediente según los comensales seleccionados por el usuario.

- **Corrección de Usabilidad:** Se detectó un detalle que afectaba la experiencia: los campos de búsqueda no se limpiaban al volver a la pantalla principal. Se utilizó el prompt:
"al hacer clic en 'volver a buscar' no se blanquean los campos filtro, hagamos que esté como si ingresara por primera vez". La IA reconoció inmediatamente la importancia de este detalle de usabilidad y ajustó el código para que el formulario se reiniciara a su estado inicial.

3.4. Fase 4: Planificación de la Escalabilidad

Para asegurar el crecimiento futuro de la aplicación, se consultó a la IA sobre la mejor forma de ampliar el listado de recetas, incluyendo ingredientes como tomate, zanahoria, carne,

pollo, entre otros. Se preguntó explícitamente si era momento de migrar a una base de datos.

La IA recomendó de forma contundente el uso de una base de datos para garantizar la

escalabilidad, el mantenimiento sencillo y la flexibilidad a futuro. La plataforma sugerida fue

Firestore de Google, debido a su generosa capa gratuita, facilidad de integración con JavaScript y su arquitectura sin servidor.

Adicionalmente, la IA proporcionó una guía detallada paso a paso para estructurar la base de datos, definiendo la colección "recetas" y los campos específicos para cada documento (receta), manteniendo la estructura de datos ya utilizada en el proyecto. Se crearon y estructuraron nuevas recetas como documentos, siguiendo este modelo.

3.5. Fase 5: Despliegue

El código fuente de la aplicación fue alojado en un repositorio de GitHub. Para el despliegue y la obtención de una URL pública, se recurrió a

GitHub Copilot. Este asistente proporcionó instrucciones claras y precisas para configurar

GitHub Pages, indicando los pasos a seguir en la configuración del repositorio, como seleccionar la rama de despliegue y la carpeta raíz, lo que permitió publicar el sitio de manera eficiente.

4. Desafíos y Soluciones

Durante el desarrollo del proyecto surgieron diversos desafíos, los cuales fueron abordados con una combinación de reformulación de prompts y dirección estratégica.

- **Respuestas inadecuadas de la IA:** En una ocasión, al solicitar la adición de nuevas funcionalidades, la respuesta de la IA no se alineaba con el resultado esperado. El desafío se solucionó no intentando corregir el resultado, sino **reformulando el prompt para ser más preciso y claro** sobre el objetivo deseado, lo que demuestra la importancia del rol del usuario como director del proyecto.
- **Limitaciones Técnicas de las Funcionalidades Sugeridas:** La idea de implementar un temporizador de cocción, aunque atractiva, fue analizada por la IA, que reveló un inconveniente técnico mayor relacionado con la ejecución de JavaScript en segundo plano en dispositivos móviles. La solución fue tomar una decisión informada basada en el análisis de la IA y **priorizar la robustez de las funcionalidades centrales**, postergando la característica para una versión futura.
- **Generación de Activos Gráficos:** Al generar imágenes para los platos con IA, se observó que estas se entregaban por defecto en formato cuadrado. El desafío se abordó ajustando el prompt para solicitar las imágenes en formato horizontal, lo que permitió obtener un resultado más adecuado para el diseño de la web.

5. Conclusión

La experiencia de desarrollar esta aplicación web ha sido una demostración fehaciente del poder de la Inteligencia Artificial Generativa como una herramienta de asistencia y aceleración en un ciclo de desarrollo tecnológico. La IA no solo actuó como un generador de código, sino como un socio estratégico en múltiples fases del proyecto.

Fue fundamental en la

fase de ideación, ayudando a refinar una idea vaga en un concepto de aplicación concreto y viable. Durante el

desarrollo, funcionó como un "copiloto de programación" que proveyó el código base y funcionalidades complejas como el escalado de ingredientes, reduciendo significativamente los tiempos de desarrollo. Además, demostró ser una potente herramienta para el

análisis técnico (como en el caso del temporizador) y la **planificación estratégica** (al recomendar y estructurar la migración a una base de datos).

El proceso reafirmó que el rol del desarrollador no es reemplazado, sino transformado. La clave del éxito residió en mantener el control como "director del proyecto", sabiendo qué pedir, cómo interpretar las respuestas de la IA, cuándo reformular las peticiones y cómo integrar las soluciones propuestas en una visión coherente y funcional. La documentación concurrente de los prompts y las decisiones tomadas fue crucial para mantener el rumbo y para la elaboración de este informe. En definitiva, el proyecto no solo cumple con los objetivos técnicos, sino que también representa una valiosa experiencia práctica sobre el paradigma de la co-creación hombre-máquina.