



UNLOCKING AI CHALLENGE 2026

De la idea a la app: **Flint + Snack**

Guía paso a paso para estudiantes: crear tus cuentas, programar con inteligencia artificial en Flint y probar tu aplicación en Snack con una estructura de código ordenada.



The School of the Future 2026 · Cognita Chile

Duración del desafío: 4 días, del 23 al 26 de noviembre de 2026

Equipos de 3 a 5 estudiantes · Modalidad remota y asincrónica

Antes de empezar

En este desafío vas a construir una app web o móvil, simulada en Snack/Expo, que haga accesible un conocimiento complejo (médico o financiero). Usarás dos herramientas en equipo:



`app.flintk12.com`

Es la inteligencia artificial que **escribe el código contigo**. Le explicas qué necesitas y te propone el código, te ayuda a corregir errores y te explica cómo funciona. **Su uso es obligatorio.**



Snack

`snack.expo.dev`

Es el **editor y compilador en línea** donde pegas el código, lo ejecutas y ves tu app funcionando en el navegador, en un emulador Android o en tu propio teléfono.

Tu ruta en 5 pasos

#	Paso	Qué lograrás
1	Crear tu cuenta en Snack	Guardar tu proyecto en la nube y compartirlo con tu equipo y el jurado.
2	Ingresar a Flint con el correo de tu colegio	Entrar a tu cuenta de Flint (ya creada) y dejarla lista para trabajar.
3	Crear el proyecto en Snack	Tener la base de la app y el código organizado (estructura recomendada u otra propia).
4	Ciclo Flint → Snack	Construir la app archivo por archivo, probando cada avance.
5	Revisar y entregar	App funcionando, bitácora al día y Reporte de Transparencia.

⚠ Regla de oro del desafío

Puedes usar la IA libremente, pero **tu equipo debe poder explicar todo lo que la IA genera**: la lógica de los prompts, las decisiones de diseño y cómo funciona el código. Nunca pegues código que no entiendes sin pedirle a Flint que te lo explique.

Tu app **no puede integrar ni operar como un chatbot**: Flint se usa solo para construirla.

1 Crea tu cuenta en Snack



Con tu cuenta en Snack podrás guardar tus proyectos, verlos desde cualquier computador y compartir un enlace público de tu app.

- 1 Abre `https://snack.expo.dev` en el navegador (se recomienda Google Chrome).
- 2 Haz clic en **Log in** (arriba a la derecha) y luego en **Sign up** para crear una cuenta nueva.
- 3 Completa el formulario: **correo electrónico** (usa el correo de tu colegio), **nombre de usuario** y **contraseña**. Para el nombre de usuario, sigue las recomendaciones del recuadro de abajo.
- 4 Revisa tu bandeja de entrada y **confirma tu correo** con el enlace que recibes.
- 5 Vuelve a `snack.expo.dev` e inicia sesión. Debes ver tu nombre de usuario en la esquina superior derecha.

Tu nombre de usuario: identificable, pero seguro

El nombre de usuario será visible para otras personas (por ejemplo, en el enlace público de tu app). Debe permitir que tu tutor y el jurado te reconozcan, **sin revelar información personal**.

✓ **Usa:** tu primer nombre + la inicial de tu apellido + un número que no sea tu fecha de nacimiento.
Ej.: `camilar27`, `tomas_g84`

✗ **No incluyas:** tu apellido completo, nombre o sigla del colegio, curso, ID (RUT, DNI, CPF), fecha de nacimiento, ciudad ni teléfono.

Opcional, pero muy recomendado: prueba en tu teléfono

- 1 En tu teléfono Android, instala la app **Expo Go** desde Google Play.
- 2 Inicia sesión en Expo Go con **la misma cuenta** que creaste.
- 3 Cuando tengas un proyecto abierto en Snack, elige la pestaña **My Device** en el panel derecho y escanea el código QR con Expo Go. Verás tu app real en el teléfono.

¿Una cuenta por estudiante o una por equipo?

Cada integrante crea su propia cuenta para poder practicar. Para el proyecto oficial, el equipo define **un único Snack principal** (guardado en la cuenta de un integrante) y comparte su enlace con todos. Así evitan tener varias versiones distintas de la misma app.

✓ Checklist del paso 1

- ☐ Tengo mi cuenta de Snack creada y el correo confirmado.
- ☐ Mi nombre de usuario es identificable, pero no incluye datos personales ni del colegio.
- ☐ Puedo iniciar sesión en `snack.expo.dev` y veo mi usuario.
- ☐ (Opcional) Tengo Expo Go instalado en un teléfono Android.

2

Ingresa a Flint con el correo de tu colegio



Tu cuenta de Flint ya fue creada: no necesitas registrarte, solo ingresar con tu correo del colegio.

- 1 Abre `https://app.flintk12.com`.
- 2 Usa tu **correo del colegio**, el que termina en el dominio de tu colegio (por ejemplo `@colegiopumahue.cl` o `@americanbritish.cl`). Escríbelo en el campo **School email** y presiona **Next**.
- 3 Ingresa tu **contraseña**: es la misma que usas para entrar a tu correo institucional del colegio.
- 4 Escribe un mensaje de prueba (por ejemplo: "Hola, ¿qué es React Native?") para confirmar que todo funciona.

⚠ Correo del colegio, no el de Cognita

No uses el dominio **@cognita.com** ni un correo personal. Si aparece un error de acceso, **no crees una cuenta con otro correo**: avisa a tu profesor tutor.

Cómo usar Flint en este desafío

Úsalo para...	Ejemplo
Generar código	"Crea la pantalla de inicio con un título y tres botones".
Corregir errores	Pegar el mensaje de error rojo de Snack y pedir la causa.
Entender el código	"Explícame línea por línea qué hace este archivo".
Diseñar la interfaz	"Propón una paleta y tamaños de letra legibles para adultos mayores".
Simplificar contenidos	"Reescribe este término médico para que lo entienda un niño de 12 años".

✓ Checklist del paso 2

- ☐ Ingresé a Flint con el correo de mi colegio (no con @cognita.com ni uno personal).
- ☐ Envié un mensaje de prueba y recibí respuesta.

3 Crea el proyecto y organiza tu código

Una app bien organizada es más fácil de construir con IA, de corregir y de explicar al jurado. Antes de pedir código, preparen el “esqueleto” del proyecto.

 La estructura que verás aquí es solo una recomendación

Es un **ejemplo opcional**: tu equipo puede **crear su propia estructura** (otras carpetas, otros nombres). Lo importante es que el código esté **ordenado** y que puedan **explicar cómo está organizado**.

- 1
- En Snack, con tu sesión iniciada, crea un proyecto nuevo (botón `New Snack` o `Create`).
- 2
- Cambia el nombre del proyecto (arriba a la izquierda) por el de tu equipo: `UnlockingAI-NombreEquipo` , y agrega una descripción breve.
- 3
- Presiona `Save` . Copia el enlace del Snack y compártelo con tu equipo y tu tutor.
- 4
- En el panel de archivos (izquierda), crea las carpetas y archivos de tu estructura (puedes usar la recomendada o la tuya). Para crear un archivo dentro de una carpeta, escribe la ruta completa, por ejemplo `screens/HomeScreen.js` . **Ojo:** Snack distingue mayúsculas y minúsculas (`homeScreen` ≠ `HomeScreen`).

Estructura recomendada ejemplo opcional

```
UnlockingAI-NombreEquipo/  
├── App.js           ← punto de entrada  
├── package.json    ← dependencias  
├── screens/        ← una pantalla por archivo  
│   ├── HomeScreen.js  
│   └── ResultScreen.js  
├── components/    ← piezas reutilizables  
│   ├── PrimaryButton.js  
│   └── InfoCard.js  
├── data/          ← contenidos y textos  
│   └── content.js  
├── theme/         ← colores y estilos  
│   └── colors.js  
├── utils/         ← funciones de lógica  
│   └── helpers.js  
└── assets/        ← imágenes e íconos
```

¿Para qué sirve cada parte?

Carpeta	Contiene
<code>App.js</code>	Arranca la app y elige la pantalla.
<code>screens/</code>	Cada pantalla de la app.
<code>components/</code>	Piezas que se repiten.
<code>data/</code>	Textos y contenidos simplificados.
<code>theme/</code>	Colores y estilos.
<code>utils/</code>	Cálculos y lógica.
<code>assets/</code>	Imágenes del proyecto.

Convenciones de nombres sugerencia

Elemento	Formato	Ejemplo
Pantallas y componentes	PascalCase + .js	<code>HomeScreen.js</code> , <code>InfoCard.js</code>
Variables y funciones	camelCase	<code>selectedTopic</code> , <code>calculateRisk()</code>
Constantes	MAYÚSCULAS	<code>MAX_QUESTIONS</code>

Repitan este ciclo para **cada archivo o mejora**. Avanzar de a poco permite detectar los errores apenas aparecen.

1. Pedir

Escribe en Flint qué archivo necesitas

2. Entender

Lee la explicación y pregunta lo que no entiendas

3. Pegar

Copia el código en el archivo correcto de Snack

4. Probar

Revisa la vista previa en Web / Android

5. Registrar

Guarda el Snack y anota en la bitácora

Paso a paso detallado

- 1 Pide un archivo concreto en Flint.** Cuéntale de qué trata tu app, indica la ruta del archivo y lo que debe hacer.
- 2 Lee la respuesta antes de copiar.** Verifica que la ruta del archivo coincida con la estructura de tu proyecto y que el código esté completo (sin "..." ni partes omitidas).
- 3 Copia el código completo** con el botón de copiar de Flint.
- 4 En Snack, abre o crea el archivo con esa ruta exacta,** selecciona todo su contenido (Ctrl + A / Cmd + A) y reemplázalo por el nuevo código (Ctrl + V / Cmd + V).
- 5 Si Flint mencionó una librería,** Snack suele mostrar un aviso para agregarla; acéptalo o revisa que aparezca en `package.json`.
- 6 Mira la vista previa** (panel derecho): pestañas **Web**, **Android** o **My Device** con Expo Go. Prueba en Web o Android antes de dar algo por terminado.
- 7 ¿Pantalla roja o error?** Copia el mensaje de error completo y llévalo a Flint junto con el código del archivo y pídele que te explique la causa. No borres código al azar.
- 8 Cuando funcione, presiona **Save**** en Snack y anota en la bitácora: prompt usado, resultado, problemas y aprendizaje.

⚠ Errores que hay que evitar

- Pegar el código en el archivo equivocado o mezclar dos archivos en uno.
- Pedir "haz toda la app" de una vez: el resultado es difícil de corregir y de explicar.
- Copiar solo una parte del código cuando Flint entregó el archivo completo.
- Trabajar varios integrantes sobre el mismo archivo al mismo tiempo: coordinen quién edita qué.
- Escribir claves, contraseñas o datos personales reales en el código.
- Pedirle a Flint que agregue un chatbot o asistente de IA a tu app: no está permitido.

5

Revisa y entrega tu proyecto

La entrega final es el **26 de noviembre**. Antes de enviar, revisen juntos este checklist.

✓ Checklist final

- ☐ La app abre y funciona en **Snack** (vista **Web** o **Android**) sin errores.
- ☐ Una persona que no conoce la app puede usarla **sin instrucciones previas**.
- ☐ La app resuelve de verdad la problemática elegida (no es solo una maqueta).
- ☐ Los contenidos están simplificados, verificados y con lenguaje claro.
- ☐ El código está **ordenado en carpetas y archivos** con una estructura clara (la recomendada u otra propia) que el equipo sabe explicar.
- ☐ El Snack está guardado, con nombre del equipo y un **enlace público** que funciona.
- ☐ Todo el equipo puede explicar cómo funciona cada archivo y los prompts principales.
- ☐ La **Bitácora diaria** está completa (objetivos, prompts, problemas y aprendizajes de cada día).
- ☐ El **Reporte de Transparencia** (400–500 palabras o video de máx. 2 min) explica la arquitectura y la lógica de prompts.
- ☐ La app **no integra ni opera a través de un chatbot** (Flint solo se usa para construirla).

Recuerda: la bitácora también se evalúa (10%)

Cada día registren qué le pidieron a Flint, qué funcionó, qué falló y qué aprendieron. Copiar los prompts reales que usaron es la mejor evidencia de un uso consciente de la IA.