GUÍA NODESTUDIO

Avatar interactivo con Claude Code

Un avatar que te mira y reacciona a lo que hacés en tu página.

⏱ ~20 minutos

1 Creá tu avatar en Avatar Studio

Es gratis, corre en el navegador y no pide cuenta.

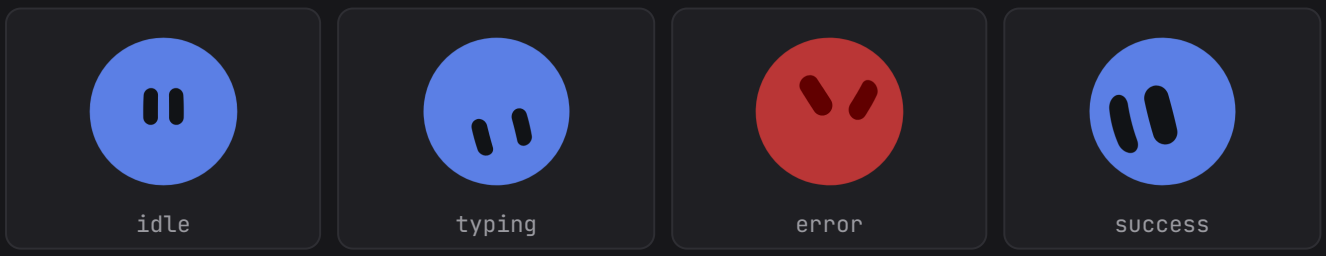
WEB avatars.bible-strong.app

Expresión = una cara: dónde miran los ojos, cómo gira la cabeza, qué color tiene.

Animación = una o varias expresiones en secuencia, con parpadeo. Tu página va a usar animaciones.

1. **Elegí o creá el avatar.** Arrancá con uno de los avatares de la galería o tocá `+`. Doble clic para editar forma, colores y ojos.
2. **Creá las caras.** Pestaña **Expressions** → **New expression**. Mové ojos y cabeza hasta lograr la cara. Podés darle un color propio (rojo para error, verde para success).
3. **Armá las animaciones.** Pestaña **Animations** → **New animation**. Completá **Semantic key**, agregá expresiones al timeline, elegí **Playback mode**, dejá **Enable blinking** prendido y tocá **Save**.

Semantic key	Cuándo aparece	Playback mode	Idea de cara
`idle`	En reposo	Loop	**Ya viene hecha**, no la tenés que crear
`typing`	Mientras escribe	Loop o Ping-pong	Mirada hacia abajo, concentrado
`error`	Algo está mal	Once	Ceño fruncido, tono rojizo
`success`	Registro OK	Once	Ojos grandes y felices, tono verde



OPCIONAL: OJOS QUE SE MUEVEN DE VERDAD

Creá 4 animaciones más de una sola expresión cada una (Loop + parpadeo): `look-left`, `look-right`, `look-up`, `look-down`. El prompt las usa para que los ojos apunten al cursor.

2 Exportalo y pasáselo a Claude Code

Exportar el `.avatar.json`

1. Pestaña **Export**. Elegí **React / TypeScript** (si no tenés proyecto, elegí este) o **JavaScript / ESM** si tu página no usa React.
2. En **Animations to export** tocá **Customize** y asegurate de que estén `idle`, `typing`, `error` y `success` (y las `look-*` si las hiciste).
3. Tocá **Download .avatar.json definition**. Se descarga un archivo tipo `strobi.avatar.json`.

> **HACÉ UN BACKUP**
>
> Avatar Studio guarda todo en tu navegador. Si borrás los datos del navegador, lo perdés. En **Export** → **Studio project** podés bajar una copia de todo tu trabajo.

Pasárselo a Claude Code

1. Creá una carpeta nueva (o abrí tu proyecto) y poné adentro el `.avatar.json`.
2. Abrí Claude Code en esa carpeta.
3. Pegá el prompt de la página siguiente. Cambiá `TU-AVATAR.avatar.json` por el nombre real de tu archivo.

Este es el repositorio oficial. No lo tenés que descargar: el link ya va adentro del prompt y Claude Code lo lee para usar las funciones que existen de verdad.

REPO github.com/smontlouis/bible-strong-avatar-lab

Paquetes que va a instalar Claude Code: `@bible-strong/avatar-react` (React) o `@bible-strong/avatar-web` (JavaScript sin React).

3 El prompt: copialo entero

PROMPT PARA PEGAR EN CLAUDE CODE

Construí una página de registro (signup) moderna donde un avatar interactivo sea el protagonista.

CONTEXTO

- Hice el avatar en Avatar Studio (https://avatars.bible-strong.app/). Repo oficial: https://github.com/smontlouis/bible-strong-avatar-lab
- Mi avatar está en esta carpeta: TU-AVATAR.avatar.json
- Antes de programar, leé el README del repo y el de packages/avatar-react (o packages/avatar-web si el proyecto no usa React). Usá solo la API documentada ahí. No inventes props ni métodos.
- Si la carpeta está vacía, creá un proyecto con Vite + React + TypeScript e instalá @bible-strong/avatar-react.
- Abrí el .avatar.json y confirmá que existan las animaciones idle, typing, error y success. Si falta alguna, avisame en lugar de inventarla.

ESTADOS DEL AVATAR

- idle: en reposo. typing: mientras el usuario escribe (volvé a idle tras ~1,2 s sin teclear). error: cuando un campo o el envío no validan. success: cuando el registro sale bien.
- Controlalo con el ref del componente (play). Llamá play solo cuando el estado cambia, no en cada tecla.
- error y success son "once": al terminar el avatar queda congelado. Usá onAnimationEnd para volver a idle (en success, después de mostrar la pantalla de éxito).

MIRADA QUE SIGUE AL CURSOR

- La librería no trae seguimiento de cursor: implementalo vos sin tocar node_modules.
- Inclinación 3D sutil del contenedor del avatar hacia el cursor (máx. ~12°) con interpolación suave en requestAnimationFrame, con inercia, nunca brusca.
- Si existen las animaciones look-left, look-right, look-up y look-down, usalas en idle para que los ojos apunten al cursor. Cambiá solo cuando cambia de zona, con margen para que no tiemble.
- Mientras escribe, que mire hacia el campo activo. En mobile, que siga al campo con foco.

DISEÑO

- Desktop en dos columnas: avatar grande (320px o más) como protagonista y formulario compacto al lado. Mobile: avatar arriba.
- Glow sutil detrás del avatar (degradado radial difuso, opacidad baja) que respira lento y se tiñe apenas de rojo en error y de verde en success.
- Que no parezca generado por IA: nada de degradados violeta-azul, glassmorphism, emojis, íconos decorativos ni una tarjeta centrada sobre fondo degradado. Paleta a partir de los colores del .avatar.json, una tipografía con carácter y mucho aire.
- Animaciones fluidas de 150 a 400 ms con easing suave. Respetá prefers-reduced-motion.

FORMULARIO

- Nombre, email, contraseña (con mostrar/ocultar), confirmar contraseña y botón "Crear cuenta". Validación en el cliente con mensajes claros, shake corto en el campo con error y botón con estado de carga. Simulá el envío con 1 s de espera (sin backend).

AL TERMINAR

- Levantá el proyecto, probá los 4 estados y decime cómo abrirlo en el navegador.

4 Si quedó básico, iterá

Pegá estos en el mismo chat de Claude Code, uno por vez. Mirá el resultado antes de pasar al siguiente.

Más vida en el movimiento

PEDÍSELO ASÍ

```
Quedó muy estático. Hacé que la inclinación hacia el cursor tenga efecto resorte, que el glow respire más lento y reaccione al estado, y que el avatar dé un pequeño rebote al pasar a success. No cambies la lógica de estados y mantené las animaciones por debajo de 400 ms.
```

Menos genérico

PEDÍSELO ASÍ

```
El diseño se ve genérico. Rehacé solo la parte visual: armá la paleta con los colores del .avatar.json, usá una tipografía con más personalidad, dale más aire y jerarquía, y sacá cualquier degradado violeta, glassmorphism, emoji o ícono decorativo. El avatar tiene que ser lo primero que se ve.
```

El detalle que enamora

PEDÍSELO ASÍ

```
Cuando el usuario entra al campo de contraseña, que el avatar haga como que no mira: reproducí la animación shy si está en el .avatar.json (si no está, avisame) y volvé a idle cuando sale del campo.
```

> **TIP**
>
> `shy` viene de fábrica en Avatar Studio junto a otras como `thinking`, `listening`, `happy`, `celebrate`, `sad` y `confused`. Para usarlas, tildalas en **Animations to export** y volvé a descargar el `.avatar.json`.

5 Lo que encontré al probarlo

EL AVATAR NO CAMBIA DE CARA

Casi siempre es la Semantic key. Sin key, la animación no se exporta. Con un typo, el avatar no hace nada y solo aparece un error en la consola. Tienen que ser exactamente `idle`, `typing`, `error` y `success`, en minúscula.

SE QUEDA TRABADO EN ERROR O SUCCESS

Las animaciones en modo Once terminan y dejan la última cara fija. El prompt ya le pide a Claude Code que vuelva a `idle` al terminar. Si te pasa, pedíle que lo revise.

EL SEGUIMIENTO DEL CURSOR NO VIENE DE FÁBRICA

Avatar Studio no tiene una función de "mirar al cursor". La construye Claude Code con el prompt. Si querés que se muevan los ojos y no solo la cabeza, creá las animaciones `look-*` del paso 1.

LICENCIA AGPL-3.0

El proyecto y sus paquetes usan esta licencia. Para aprender o para tu propia web, sin drama. Si lo vas a meter en un producto comercial o en un proyecto de un cliente, leé antes qué te exige.

¿Dónde más usarlo?

Onboarding

Que acompañe cada paso y festeje con `celebrate` al terminar.

Chat con IA

`listening` mientras escribís, `thinking` mientras responde.

Página 404

`confused` o `sad`, siguiendo el cursor para que el error no frustre.



NodeStudio

Automatizamos procesos de negocio con IA y sistemas a medida.

nodestudioai.com