



AULAFY

MANUAL PRÁCTICO · DE CERO A PROYECTO

Codex desde cero

Trabaja con archivos, automatiza tareas
y construye proyectos

20 lecciones · Ejemplos resueltos · Casos integrales · Paquete de práctica
Nivel cero absoluto a intermedio · Puente técnico opcional

Codex desde cero

Trabaja con archivos, automatiza tareas y construye proyectos

Versión del contenido: 2, ampliada y aprobada.

Revisión documental: 14 de julio de 2026.

Nivel: cero absoluto a intermedio, con puente técnico opcional.

Duración orientativa: 14-20 horas más proyecto final.

Plataformas: macOS, Windows y Linux según la superficie utilizada.

Este manual enseña a utilizar Codex con alcance limitado, revisión humana y evidencia. Las funciones de producto, disponibilidad e interfaces pueden cambiar. Consulta las fuentes oficiales enlazadas y la fecha de revisión antes de impartir o publicar una nueva edición.

Los nombres, correos y situaciones del paquete de práctica son ficticios. No utilices información personal ni originales irremplazables durante el aprendizaje.

Cómo aprender con este manual

Cada práctica utiliza el mismo ciclo. Si falta un paso, el resultado todavía no está listo.



PRÁCTICA

Actividad para realizar, con archivos y resultado observable.

ALARMA

Situación en la que debes detenerte y revisar antes de continuar.

TERMINADO

Evidencia mínima para considerar completada una tarea.

COMPROBACIÓN

Control independiente que no depende de aceptar la respuesta del agente.

Consejo editorial: empieza con la ruta esencial si es tu primer contacto. Completa la ruta general para usar Codex en tareas cotidianas y añade la ruta técnica cuando necesites repositorios, CLI o IDE.

Mapa del manual

Presentación del curso	1
Módulo 1. Orientarse antes de actuar	3
Lección 1. Qué es Codex y qué no es	4
Lección 2. Dónde utilizar Codex: app, web, CLI e IDE	7
Lección 3. Instala e inicia sesión en tu plataforma	10
Módulo 2. Tu primer resultado seguro	14
Lección 4. Crea una carpeta de práctica y empieza en solo lectura	15
Lección 5. Pide resultados claros sin aprender «prompt engineering»	18
Lección 6. Planifica, corrige el rumbo y cierra con evidencia	21
Módulo 3. Trabajo cotidiano sin programar	24
Lección 7. Organiza archivos sin perder los originales	25
Lección 8. Crea documentos, datos, presentaciones, PDF e imágenes	28
Lección 9. Investiga en la web y utiliza el navegador con criterio	31
Lección 10. Usa Computer Use sin entregar el control de tu equipo	34
Módulo 4. Construir sin saber programar	37
Lección 11. Construye tu primera web local	38
Lección 12. Automatiza una tarea repetitiva con vista previa	41
Módulo 5. Organización, control y recuperación	44
Lección 13. Organiza proyectos, tareas y entornos	45
Lección 14. Permisos, privacidad, secretos y acciones externas	48
Lección 15. Copias de seguridad y Git sin miedo	52
Módulo 6. Personalizar y reutilizar	55
Lección 16. Instrucciones personales y AGENTS.md	56
Lección 17. Elige entre skills, plugins y MCP	59
Lección 18. Programa tareas que puedas revisar	63
Módulo 7. Puente técnico y proyecto final	66
Lección 19. Da el salto a CLI, IDE y repositorios cuando lo necesites	67
Lección 20. Proyecto final según tu perfil	71
Apéndice A. Matriz rápida por plataforma	75
Apéndice B. Plantillas de prompts	76
Solo lectura	76
Plan antes de actuar	76
Crear sin sobrescribir	76
Investigación	76
Corregir el rumbo	76
Cierre con evidencia	77
Borrador sin enviar	77
Apéndice C. Solución de problemas	78

No puedo iniciar sesión	78
Codex no ve la carpeta	78
Puede leer, pero no escribir	78
Pide red repetidamente	78
Browser, Computer Use, plugins o Scheduled no aparecen	78
Linux avisa del sandbox	78
Windows mezcla PowerShell y WSL	78
Computer Use controla otra ventana	79
No hay evidencia	79
Cambió demasiados archivos	79
La tarea no termina	79
Diagnóstico técnico	79
Apéndice D. Glosario	80
Apéndice E. Plan de capturas para la futura edición	81
Apéndice F. Fuentes y revisión	82
Fuentes oficiales principales	82
Revisión antes de publicar	82
Datos deliberadamente no fijados	82
Apéndice G. Paquete de práctica reproducible	83
Archivo ideas.txt	83
Archivo tareas.txt	83
Archivo notas.txt	84
Archivo seguimiento.txt	84
Archivo clientes-con-incidencias.csv	84
Archivo clientes-semana-2.csv	84
Archivo brief.md	85
Archivo recursos/textos.md	85
Archivo preguntas.md	85
Control del paquete antes de empezar	86
Apéndice H. Cuatro casos integrales	87
Caso 1. Sol prepara una unidad didáctica	87
Caso 2. Marta prepara el seguimiento semanal	87
Caso 3. Leo redacta una comparativa con fuentes	88
Caso 4. Dani corrige una incidencia pequeña	88
Cómo realizar los casos en grupo	89
Apéndice I. Clínica de prompts	90
Ejercicio 1. Diagnostica antes de reescribir	91
Ejercicio 2. Reduce sin perder control	91
Ejercicio 3. Añade una pausa inteligente	91
Plantilla universal mínima	91
Apéndice J. Autoevaluación y guía para facilitar el curso	92
Autoevaluación inicial	92
Evaluación final por demostración	92
Rúbrica común de 0 a 3	92
Guía para una persona facilitadora	93
Adaptaciones de accesibilidad	93
Preguntas para cerrar cada módulo	93
Lista final de aprobación editorial	94

Presentación del curso

Codex es un agente de inteligencia artificial de OpenAI capaz de trabajar con archivos, ejecutar herramientas, comprobar resultados y realizar tareas de software. No necesitas saber programar para aprender a dirigirlo, revisar lo que hace y construir proyectos pequeños. Sí necesitas comprender tres cosas: qué herramienta elegir, qué acceso estás concediendo y cómo comprobar el resultado antes de utilizarlo.

Este curso empieza desde cero. Primero aprenderás a trabajar en una carpeta de práctica sin poner en riesgo tus documentos. Después utilizarás lenguaje natural para organizar información, crear archivos, investigar, revisar páginas web y automatizar tareas. Las últimas lecciones ofrecen un puente opcional hacia la terminal, el IDE y el curso «Codex para programadores».

El objetivo no es que aceptes todo lo que produce una IA. El objetivo es que puedas encargar un resultado, observar el proceso, detectar límites, pedir correcciones y conservar evidencia de lo que realmente se ha comprobado.

Al terminar serás capaz de

- Elegir entre Chat, ChatGPT Work y Codex según el resultado que necesitas.
- Utilizar la aplicación de escritorio en macOS o Windows, y Codex CLI o la extensión de IDE en Linux.
- Dar acceso únicamente a una carpeta de trabajo y comprender los permisos básicos.
- Escribir encargos claros sin aprender una sintaxis especial.
- Trabajar con archivos, carpetas, capturas, datos y páginas web.
- Utilizar el navegador y Computer Use con límites seguros.
- Construir una web sencilla sin escribir el código manualmente.
- Automatizar una tarea repetitiva con vista previa y recuperación.
- Organizar proyectos y tareas sin mezclar contextos.
- Crear instrucciones reutilizables y elegir entre skills, plugins y MCP.
- Programar tareas que produzcan resultados revisables.
- Decidir cuándo avanzar hacia Git, CLI, IDE y flujos profesionales.

Requisitos reales

- Poder crear una carpeta y copiar archivos.
- Una cuenta de ChatGPT con acceso a las funciones que quieras utilizar.
- Conexión a internet para iniciar sesión y utilizar el servicio.
- En macOS o Windows, la aplicación de escritorio de ChatGPT.
- En Linux, un navegador para ChatGPT Work y, para Codex local, una terminal o un editor compatible.

No necesitas Git, un repositorio, una clave API ni experiencia programando.

Cómo utilizar el curso

Las lecciones 1 a 16 forman la ruta general. La 17 presenta funciones reutilizables y la 18 enseña tareas programadas. La 19 es un puente técnico opcional. La 20 ofrece proyectos finales según tu perfil.

Cada práctica utiliza una carpeta o archivos de entrenamiento. No trabajes con originales irremplazables durante el aprendizaje. Si una función no aparece en tu cuenta, lee la explicación, completa la alternativa indicada y consulta la fuente oficial de disponibilidad.

Qué incorpora esta versión ampliada

Esta versión no añade dificultad por añadir páginas. Añade apoyos para que una misma explicación sirva a personas con experiencias muy distintas:

- ejemplos completos, desde el encargo inicial hasta la comprobación final;
- versiones deficientes y mejoradas de un mismo prompt;
- alternativas para estudio, docencia, trabajo autónomo, creación y desarrollo;
- preguntas de autoevaluación con respuesta razonada;
- archivos de práctica que se pueden copiar literalmente;
- casos integrales para unir varias lecciones;
- una clínica de prompts para aprender a corregir encargos ambiguos;
- criterios observables para saber si una práctica está terminada.

Tres rutas posibles

No todas las personas tienen que recorrer el curso de la misma manera.

Ruta esencial - 5 a 7 horas. Lecciones 1, 3, 4, 5, 6, 7, 13, 14 y 20. Enseña a utilizar Codex con seguridad, formular encargos claros, revisar cambios y terminar un proyecto pequeño.

Ruta general - 14 a 20 horas. Lecciones 1 a 18 y un proyecto final. Es la ruta recomendada para quien empieza desde cero y quiere aplicar Codex a tareas reales sin depender de conocimientos técnicos.

Ruta técnica - 18 a 26 horas. Curso completo, incluida la lección 19, las prácticas de Git y el proyecto técnico. Prepara el salto hacia repositorios, terminal e IDE.

Los cuatro personajes guía

Los ejemplos siguen a cuatro personas ficticias. No representan categorías cerradas; sirven para mostrar cómo cambia un buen encargo según el contexto.

- **Sol, estudiante y docente:** prepara materiales, ordena apuntes y diseña actividades.
- **Marta, profesional autónoma:** organiza clientes, crea una web y automatiza seguimiento administrativo.
- **Leo, creador e investigador:** compara fuentes, prepara contenidos y conserva trazabilidad.
- **Dani, desarrollador principiante:** revisa un repositorio, modifica código y ejecuta comprobaciones.

Método de estudio: encargar, observar, comprobar, explicar

En cada práctica sigue siempre el mismo ciclo:

1. **Encargar:** describe el resultado, el contexto, los límites y la comprobación.
2. **Observar:** mira el plan, los archivos consultados, las herramientas y las solicitudes de permiso.
3. **Comprobar:** abre el resultado, compara los cambios y repite una verificación independiente.
4. **Explicar:** resume con tus propias palabras qué cambió, qué se comprobó y qué queda pendiente.

Si no puedes explicar el resultado, todavía no está listo para un uso importante.

Convenciones de las prácticas

- **Hazlo ahora:** actividad guiada que conviene ejecutar.
- **Ejemplo resuelto:** una solución posible, no una fórmula obligatoria.
- **Cambia una variable:** pequeña variación para evitar el aprendizaje mecánico.
- **Señal de alarma:** situación en la que debes detenerte y revisar.
- **Criterio de terminado:** evidencia mínima para considerar completada la actividad.
- **Comprueba que lo has entendido:** preguntas con solución desplegable.

Módulo 1. Orientarse antes de actuar

Lección 1. Qué es Codex y qué no es

Tiempo: 25 minutos

Superficie: cualquier superficie de ChatGPT

Plataformas: macOS, Windows y Linux

Qué vas a aprender

- Qué significa que Codex sea un agente.
- La diferencia entre responder, crear un entregable y actuar sobre un proyecto.
- Cuándo no debes utilizar Codex.

Una explicación sin jerga

Un chatbot contesta a mensajes. Un agente puede además observar un entorno, utilizar herramientas y avanzar durante varios pasos hacia un resultado. En una tarea local, Codex puede leer los archivos de la carpeta elegida, modificarlos si tiene permiso y ejecutar programas para comprobar su trabajo.

Eso no convierte a Codex en una autoridad. Puede interpretar mal una petición, utilizar una fuente incorrecta, cambiar más de lo necesario o presentar como comprobado algo que no llegó a ejecutar. Tu papel consiste en definir el resultado, limitar el alcance y revisar la evidencia.

El mapa que evita la mayoría de confusiones

- **Chat:** preguntas, explicaciones, ideas, búsqueda y conversaciones rápidas.
- **ChatGPT Work:** investigación y entregables revisables, como documentos, presentaciones, hojas de cálculo, PDF o Sites.
- **Codex:** software y tareas técnicas que necesitan contexto de proyecto, archivos, comandos, pruebas o revisión de cambios.

Una persona no programadora puede utilizar Codex para construir una web pequeña, transformar archivos o crear una herramienta local. Sin embargo, si el resultado principal es un informe, una presentación o una hoja de cálculo, empieza por Work. Elegir la superficie correcta forma parte de hacer bien la tarea.

Tres preguntas antes de empezar

1. ¿Necesito una respuesta o conversación? Usa Chat.
2. ¿Necesito un archivo final visual que revisar? Usa Work.
3. ¿Necesito que el agente trabaje dentro de una carpeta o proyecto técnico? Usa Codex.

Práctica: clasifica tareas

Elige Chat, Work o Codex:

1. «Explicame la inflación con un ejemplo familiar».
2. «Crea una presentación de diez diapositivas a partir de estas notas».
3. «Revisa esta carpeta y construye una web para presentar mi servicio».
4. «Compara información actual de tres páginas y cita fuentes».
5. «Corrige el error de este proyecto y ejecuta sus pruebas».

Solución razonada: 1 Chat; 2 Work; 3 Codex; 4 Chat o Work según si quieres conversación o informe; 5 Codex.

Encargo listo para copiar

Quiero conseguir este resultado: [describelo]. Antes de empezar, dime si debo usar Chat, ChatGPT Work o Codex. Explica la elección en lenguaje sencillo, qué acceso necesitarías y qué tendría que revisar yo al final. No realices ninguna acción todavía.

Resultado esperado

Una recomendación de superficie, una explicación del acceso necesario y una lista de comprobaciones. La respuesta no debe iniciar cambios.

Comprobación y recuperación

- Confirma que la recomendación responde a tu resultado y no solo a la herramienta disponible.
- Si propone actuar, corrige: «Solo quiero orientación; no hagas cambios».
- No hay nada que deshacer porque no se ha concedido acceso ni autorizado una acción.

Error frecuente

Llamar Codex a todo. Esto lleva a utilizar una superficie técnica para un trabajo que Work puede presentar y revisar mejor.

Evidencia que debes guardar

Escribe tres tareas tuyas y anota al lado qué superficie utilizarías y por qué.

Ejemplo resuelto: una petición, tres destinos posibles

Sol escribe: «Tengo treinta páginas de apuntes y quiero preparar el examen». La petición todavía no permite elegir bien la herramienta. Al concretarla aparecen tres trabajos diferentes:

- «Explicame este concepto con una analogía y hazme cinco preguntas» es una conversación: **Chat**.
- «Convierte estos apuntes en una guía de estudio de ocho páginas, con portada y tablas» pide un entregable visual: **Work**.
- «Lee los archivos de esta carpeta, detecta cuáles están duplicados y crea un pequeño buscador local» requiere actuar sobre un proyecto y comprobarlo: **Codex**.

El aprendizaje importante no es memorizar tres marcas. Es identificar el verbo principal: **explicar**, **entregar** o **actuar**.

Hazlo ahora. Escribe tres tareas reales que tengas pendientes. Para cada una, completa esta frase:

Necesito ; el resultado será ; lo consideraré correcto cuando ____.

Si el resultado es solamente una respuesta, empieza por Chat. Si es un documento visual, empieza por Work. Si exige modificar o probar archivos de un proyecto, empieza por Codex.

Señal de alarma: estás a punto de dar acceso a una carpeta completa para resolver una pregunta que podías formular sin compartir archivos.

Criterio de terminado: puedes justificar la herramienta elegida con el tipo de resultado, no con «porque parece más potente».

Cambia una variable

- **Marta:** «Quiero saber cuánto facturé» puede ser una pregunta sobre una cifra o la creación de un sistema de seguimiento. El destino cambia al cambiar el resultado.
- **Leo:** resumir un artículo no es lo mismo que construir un archivo de investigación con fuentes y trazabilidad.
- **Dani:** pedir una explicación de un error no equivale a autorizar cambios en el repositorio.

Comprueba que lo has entendido

1. ¿Qué superficie elegirías para explicar una fórmula sin tocar archivos?
2. ¿Codex es automáticamente la mejor opción porque puede realizar más acciones?

Solución comentada

1. Chat, porque el resultado es una explicación conversacional.
2. No. Una capacidad innecesaria añade complejidad y acceso sin mejorar el resultado. Se elige por el trabajo que debe producirse.

Fuentes oficiales

- [Use ChatGPT](#)
- [Quickstart](#)
- [Projects, chats, and tasks](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 2. Dónde utilizar Codex: app, web, CLI e IDE

Tiempo: 35 minutos

Superficie: mapa comparativo

Plataformas: macOS, Windows y Linux

Qué vas a aprender

- Qué superficies existen y qué puede hacer cada una.
- Qué opción es más cómoda para empezar en tu sistema.
- Qué funciones no están disponibles en todas las superficies.

Una aplicación, varios lugares de trabajo

Desde julio de 2026, Codex forma parte de la aplicación de escritorio de ChatGPT en macOS y Windows. Dentro de la aplicación puedes elegir Codex para tareas de software o proyectos locales, junto a Chat y Work. La expresión más precisa para este curso será «**ChatGPT para escritorio, en modo Codex**».

También existen:

- **Codex en la web:** apropiado para tareas en la nube y repositorios conectados, cuando tu cuenta dispone de esa función.
- **Codex CLI:** interfaz dentro de la terminal. Es local, directa y automatizable.
- **Extensión de IDE:** Codex dentro de editores compatibles con Visual Studio Code.
- **ChatGPT Work en la web:** la alternativa accesible para Linux cuando el trabajo consiste en archivos subidos, investigación o entregables y no necesita actuar sobre una carpeta local.

Matriz de decisión

Necesidad	Superficie recomendada	Motivo
Empezar sin terminal en macOS/Windows	Aplicación de escritorio	Proyectos, previsualización, permisos y revisión visual en una interfaz
Crear documentos, hojas, presentaciones o PDF	Work en web o escritorio	Previsualización y revisión de entregables
Trabajar sobre archivos locales en Linux	CLI o extensión de IDE	No hay aplicación de escritorio para Linux
Trabajar junto al código abierto en el editor	Extensión de IDE	Selección y archivos del editor como contexto
Automatizar un análisis repetible	CLI con <code>codex exec</code>	Salida y ejecución no interactiva
Navegar e interactuar con páginas	Web o aplicación de escritorio	Browser no está disponible en CLI ni en la extensión
Controlar aplicaciones de escritorio	Aplicación en macOS/Windows	Computer Use necesita acceso a la interfaz gráfica

Local, worktree y nube

En la aplicación, una tarea de Codex puede ejecutarse:

- **Local:** sobre la carpeta actual de tu ordenador.
- **Worktree:** en una copia de trabajo aislada de un repositorio Git.
- **Cloud:** en un entorno remoto configurado.

Si todavía no sabes qué significa worktree o repositorio, elige Local dentro de la carpeta de práctica del curso. Más adelante aprenderás cuándo conviene aislar cambios.

Disponibilidad variable

Algunas funciones dependen del plan, la cuenta, la región, la organización o un despliegue gradual. La señal fiable no es una captura de este curso, sino que la opción aparezca en tu interfaz y esté permitida por tu espacio de trabajo. No cambies de plan únicamente porque una lección muestre una función opcional; utiliza la alternativa indicada.

Práctica: elige tu ruta inicial

Completa una frase:

- Uso macOS o Windows y quiero evitar la terminal: empezaré en _____.
- Uso Linux y necesito actuar sobre una carpeta local: empezaré en _____.
- Solo quiero crear un informe a partir de archivos subidos: empezaré en _____.

Solución: aplicación de escritorio en modo Codex; CLI o extensión de IDE; ChatGPT Work.

Encargo listo para copiar

Mi sistema operativo es [macOS/Windows/Linux]. Quiero [resultado] y prefiero [usar/no usar] la terminal. Recomiéndame una superficie de ChatGPT o Codex. Incluye una alternativa si la función principal no está disponible en mi cuenta. No me des todavía pasos de instalación.

Comprobación y recuperación

La recomendación debe indicar sistema, superficie, tipo de acceso y alternativa. Si mezcla instrucciones de Windows y Linux, pide que las separe antes de continuar.

Error frecuente

Seguir una captura antigua en lugar de la interfaz actual. Los nombres y ubicaciones cambian. Busca la función por su nombre y consulta la documentación enlazada si no aparece.

Evidencia que debes guardar

- Sistema operativo y superficie elegida.
- Motivo de la elección.
- Alternativa prevista si una función no está disponible.

Ejemplo resuelto: elegir según el equipo y el momento

Marta utiliza un portátil con Windows en su oficina, un navegador en un ordenador prestado y, ocasionalmente, su teléfono. Quiere construir una web local para su negocio.

Su decisión razonada es:

1. En su portátil abre la aplicación de escritorio y selecciona únicamente la carpeta de práctica: puede trabajar con archivos y revisar el resultado localmente.
2. En el ordenador prestado no inicia un proyecto con documentos sensibles; utiliza Chat o Work para una consulta sin datos privados.
3. En el teléfono revisa una explicación, pero pospone la edición del proyecto hasta volver al equipo que contiene la carpeta.

El mejor entorno no es siempre el más avanzado. Es aquel en el que puedes controlar el contexto, observar las acciones y comprobar el resultado.

Miniárbol de decisión: si necesitas una interfaz visual y usas macOS o Windows, comienza con la app. Si trabajas en Linux y quieres operar localmente, considera CLI o IDE cuando estés preparado. Si solo necesitas investigación o un entregable de Work, el navegador puede ser suficiente.

Señal de alarma: cambias de equipo y das por supuesto que el nuevo entorno tiene los mismos archivos, permisos, sesión y herramientas.

Criterio de terminado: sabes decir dónde están físicamente los archivos y en qué superficie se ejecutará el trabajo.

Cambia una variable

- Si Marta solo desea una presentación comercial, empieza en Work.
- Si Dani necesita revisar cada línea mientras programa, el IDE puede resultar más cómodo.
- Si Leo trabaja en un equipo sin sus archivos, puede investigar en la web y guardar el trabajo local para más tarde.

Comprueba que lo has entendido

1. ¿La misma tarea puede requerir una superficie distinta al cambiar de dispositivo?
2. ¿Qué pregunta debes responder antes de permitir cambios locales?

Solución comentada

Sí. Influyen el sistema operativo, la disponibilidad, la ubicación de los archivos y el nivel de control. Antes de permitir cambios debes saber qué carpeta o proyecto se ha seleccionado y qué contiene.

Fuentes oficiales

- [Quickstart](#)
- [Codex environments](#)
- [Browser](#)
- [Codex IDE extension](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 3. Instala e inicia sesión en tu plataforma

Tiempo: 30-60 minutos

Superficie: aplicación, CLI o IDE

Plataformas: rutas separadas para macOS, Windows y Linux

Qué vas a aprender

- Instalar la superficie adecuada sin mezclar instrucciones de sistemas distintos.
- Iniciar sesión con ChatGPT.
- Entender por qué una clave API no es necesaria para empezar.

Antes de instalar

Utiliza únicamente páginas y comandos oficiales. No descargues instaladores desde anuncios, repositorios desconocidos o páginas que pidan tu contraseña de ChatGPT. Comprueba que estás en chatgpt.com, openai.com o la documentación oficial enlazada.

Para aprender, la ruta recomendada es **iniciar sesión con ChatGPT**. La clave API es una forma de acceso de pago por uso para flujos locales o programáticos. No concede automáticamente las funciones de la cuenta de ChatGPT y no debe pegarse en prompts, documentos ni capturas.

Ruta A - macOS

1. Abre el [Quickstart oficial](#).
2. Descarga la aplicación de escritorio indicada para macOS.
3. Mueve o instala la aplicación siguiendo el asistente del sistema.
4. Ábrela y selecciona **Continue to sign in**.
5. Completa el inicio de sesión en la ventana del navegador.
6. Vuelve a la aplicación y confirma que aparece tu cuenta o espacio de trabajo.
7. En el selector de ChatGPT, localiza **Codex**.

No concedas todavía acceso a Documentos, Escritorio ni a una carpeta con archivos reales. La siguiente lección creará un espacio de práctica.

Ruta B - Windows

1. Abre la página oficial de la [aplicación para Windows](#).
2. Instálala desde Microsoft Store.
3. Como alternativa oficial desde PowerShell, puedes utilizar:

```
winget install Codex -s msstore
```

4. Abre la aplicación y completa **Continue to sign in**.
5. Confirma que aparece tu cuenta y que puedes seleccionar Codex.
6. Mantén el agente en Windows nativo. WSL2 es una opción técnica, no un requisito para empezar.

Si tu organización administra el equipo, Microsoft Store, plugins o Codex pueden estar restringidos. En ese caso no intentes eludir la política: consulta a la persona administradora.

Ruta C - Linux

La aplicación de escritorio no está disponible para Linux. Puedes utilizar ChatGPT Work en el navegador para archivos subidos y entregables. Para trabajar con una carpeta local mediante Codex, utiliza la CLI o la extensión de IDE.

En una distribución compatible, abre una terminal y ejecuta el instalador autónomo oficial:

```
curl -fsSL https://chatgpt.com/codex/install.sh | sh
codex
```

Sigue el inicio de sesión que se abre en el navegador. Para que el sandbox de Linux sea fiable, instala bubblewrap con el gestor de paquetes de tu distribución:

```
# Ubuntu o Debian
sudo apt install bubblewrap

# Fedora
sudo dnf install bubblewrap
```

Si no tienes permisos para instalar paquetes, utiliza ChatGPT Work en la web o pide ayuda a la persona administradora. No desactives las protecciones del sistema únicamente para completar el curso.

CLI opcional en macOS o Windows

La CLI se enseñará en la lección 19. Si ya quieres disponer de ella, la documentación oficial publica instaladores autónomos para shell (`install.sh`) y PowerShell (`install.ps1`). No necesitas instalar Node.js ni npm para seguir la ruta principal de este curso.

Comprobación de inicio de sesión

- En la aplicación o extensión, abre el menú de perfil y comprueba la cuenta activa.
- En CLI, si Codex se inicia sin pedir de nuevo credenciales y muestra el compositor, la sesión está disponible.
- No publiques capturas que muestren correo, nombre del espacio de trabajo, claves o identificadores privados.

Encargo listo para copiar

Después de iniciar sesión, escribe:

```
Estoy aprendiendo. No abras carpetas, no ejecutes comandos y no cambies archivos.
Indícame qué superficie estoy usando, qué cuenta o espacio de trabajo está activo
sin mostrar datos sensibles, y dónde puedo revisar los permisos antes de una tarea.
```

Resultado esperado

Codex identifica la superficie y señala el control de permisos. No debe solicitar acceso a una carpeta ni mostrar credenciales.

Recuperación

- Si has iniciado sesión con la cuenta incorrecta, utiliza **Log out** y vuelve a entrar.
- Si pegaste accidentalmente una clave API en un prompt, revócala inmediatamente desde el panel de API y crea otra solo si realmente la necesitas.
- Si el instalador no procede de una página oficial, cancela la instalación y elimina el archivo descargado.

Error frecuente

Confundir la suscripción de ChatGPT con la facturación de API. Son rutas de acceso distintas. Iniciar sesión con una clave API utiliza facturación de la plataforma API y puede limitar funciones dependientes de ChatGPT o de la nube.

Evidencia que debes guardar

Anota sistema operativo, superficie instalada y método de acceso. No guardes tokens, contraseñas ni claves.

Ejemplo resuelto: instalar sin adivinar

Dani instala Codex, inicia sesión y escribe inmediatamente «arregla mi proyecto». Ese salto oculta demasiadas incógnitas. Una primera comprobación más segura es:

```
Todavía no cambies nada. Indícame:
1. qué carpeta tienes abierta;
2. qué archivos de instrucciones detectas;
3. qué herramientas puedes utilizar;
4. qué limitaciones o permisos ves.
No leas archivos fuera de la carpeta seleccionada.
```

La respuesta esperada no es idéntica en todos los equipos. Debe permitir comprobar la ruta, el alcance y las capacidades. Si menciona otra carpeta, Dani detiene la tarea y corrige el entorno antes de continuar.

Después realiza una prueba mínima: crea una carpeta vacía llamada `practica-codex`, la selecciona y pide una inspección sin modificaciones. Solo cuando la ubicación es correcta continúa con la lección 4.

Diagnóstico por síntomas:

- No aparece una función: comprueba plataforma, aplicación actualizada, plan y despliegue; no inventes un botón.
- La sesión no se inicia: revisa conexión, cuenta y método de acceso antes de reinstalar.
- Codex ve la carpeta equivocada: cierra o cambia el proyecto; no intentes compensarlo con un prompt.
- En Linux buscas la misma interfaz de escritorio: utiliza la alternativa documentada para CLI, IDE o web.

Criterio de terminado: puedes abrir un entorno de práctica, identificar su ruta y recibir una descripción sin que se modifique ningún archivo.

Cambia una variable

Repite la comprobación después de cambiar de carpeta. Debe cambiar la ruta informada, pero mantenerse la instrucción de no modificar.

Comprueba que lo has entendido

1. ¿Por qué «la instalación se abrió» no demuestra que el entorno esté bien configurado?
2. Si falta una función, ¿debes seguir una captura antigua o verificar disponibilidad?

Solución comentada

Abrir la aplicación no confirma la carpeta, permisos ni herramientas disponibles. Cuando una función falta, se verifica su disponibilidad actual y se utiliza la alternativa del curso; no se fuerza una interfaz que puede haber cambiado.

Fuentes oficiales

- [Quickstart](#)
- [Authentication](#)
- [Windows app](#)
- [Sandbox](#)

Revisado contra la documentación: 14 de julio de 2026.

Prueba multiplataforma pendiente antes de publicación: instalador, inicio de sesión y nombres de menú en macOS, Windows nativo y Linux.

Módulo 2. Tu primer resultado seguro

Lección 4. Crea una carpeta de práctica y empieza en solo lectura

Tiempo: 35 minutos

Superficie: aplicación en macOS/Windows; CLI o IDE en Linux

Riesgo: bajo

Qué vas a aprender

- Elegir exactamente qué carpeta puede ver Codex.
- Diferenciar leer de modificar.
- Completar una primera tarea sin cambiar ningún archivo.

Prepara el espacio

1. Crea una carpeta llamada Aprender-Codex.
2. Dentro, crea entrada, salida y copia-seguridad.
3. Crea en entrada tres archivos de texto:
 - ideas.txt: tres ideas desordenadas.
 - tareas.txt: cinco tareas sin ordenar.
 - notas.txt: un párrafo de cualquier tema.
4. Copia la carpeta entrada dentro de copia-seguridad.

No utilices información personal. Estos archivos son material desechable de entrenamiento.

Abre únicamente la carpeta de práctica

Aplicación: crea o añade un proyecto local y selecciona Aprender-Codex. Inicia una tarea en modo Codex y revisa el control de permisos bajo el compositor.

CLI: abre una terminal, entra en la carpeta y ejecuta Codex:

```
cd /ruta/a/Aprender-Codex
codex
```

En CLI puedes utilizar `/permissions` para seleccionar un perfil de solo lectura. Si la superficie no muestra ese comando, pide una tarea de análisis y mantén el permiso de escritura desactivado en la interfaz.

Primera petición

Trabaja únicamente dentro de esta carpeta. No modifiques, renombres, muevas ni crees archivos. Lee los tres archivos de entrada y devuelve:

1. una frase sobre el contenido de cada archivo;
 2. cualquier dato que parezca personal o sensible;
 3. qué archivos no has podido leer.
- Separa hechos observados de inferencias.

Qué debes observar

Codex debería mencionar los tres archivos, resumirlos y declarar cualquier límite. No debería aparecer ningún archivo nuevo en salida ni cambiar la fecha de modificación de los originales.

Comprobación manual

1. Abre entrada con el explorador de archivos.
2. Confirma que siguen existiendo los mismos tres archivos.
3. Comprueba que salida continúa vacía.
4. Abre uno de los textos y confirma que conserva su contenido.

No necesitas confiar en la frase «no cambié nada». Comprueba el estado visible.

Recuperación

Si algo cambió, cierra la tarea, compara con copia-seguridad y restaura únicamente el archivo afectado. Conserva ambos hasta entender qué ocurrió.

Error frecuente

Abrir Documentos o la carpeta personal completa. El agente recibe demasiado contexto y una frontera poco clara. Crea un proyecto específico para cada cuerpo de trabajo.

Mini ejercicio

Repite la tarea limitando el análisis a `tareas.txt` y pide que explique qué archivos decidió no leer.

Solución comentada

No cambies archivos. Lee únicamente entrada/tareas.txt. Clasifica las tareas por tema en tu respuesta y enumera los archivos que has dejado fuera por mi restricción.

Evidencia que debes guardar

- Captura o anotación del permiso utilizado.
- Lista inicial y final de archivos.
- Resumen de Codex con sus límites.

Ejemplo resuelto: la primera inspección controlada

Sol crea practica-codex con estos cuatro archivos: `ideas.txt`, `tareas.txt`, `notas.txt` y `seguimiento.txt`. Antes de pedir una reorganización utiliza un encargo de solo lectura:

Trabaja únicamente dentro de esta carpeta.
No crees, edites, muevas ni borres archivos.
Lee los cuatro archivos de texto y devuelve:
– una tabla con nombre, propósito probable y problemas detectados;
– las contradicciones o datos incompletos;
– un plan de cambios propuesto, sin ejecutarlo.
Termina confirmando explícitamente que no modificaste nada.

Codex propone que `ideas.txt` contiene posibles actividades, `tareas.txt` mezcla fechas y prioridades, `notas.txt` guarda observaciones y `seguimiento.txt` registra el estado. Sol no acepta esa interpretación automáticamente: abre los archivos y compara dos elementos concretos de cada uno.

Luego pregunta: «¿Qué frase exacta del archivo te hizo interpretar esto como una fecha?». Esa pregunta enseña a exigir trazabilidad, no solamente una respuesta convincente.

Prueba de integridad: antes y después de la inspección, cuenta los archivos y mira sus fechas de modificación. Deben permanecer iguales. No hace falta una herramienta técnica si el explorador de archivos permite verlo.

Señal de alarma: la respuesta anuncia que «ha mejorado» o «ha corregido» contenido cuando solo se autorizó lectura.

Criterio de terminado: existe un plan concreto, puedes relacionarlo con el contenido real y los cuatro originales siguen intactos.

Cambia una variable

Añade un quinto archivo con un nombre ambiguo, como `final-final.txt`. Pide que Codex formule preguntas antes de decidir su propósito.

Comprueba que lo has entendido

1. ¿Un modo de solo lectura elimina la necesidad de revisar la respuesta?
2. ¿Qué evidencia sencilla permite detectar un cambio inesperado?

Solución comentada

No. La interpretación del contenido puede ser incorrecta aunque no se modifique nada. Puedes comparar el número de archivos, sus nombres, fechas de modificación y contenido; en proyectos con Git, también el estado y el diff.

Fuentes oficiales

- [Projects, chats, and tasks](#)
- [Sandbox](#)
- [Agent approvals & security](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 5. Pide resultados claros sin aprender «prompt engineering»

Tiempo: 40 minutos

Superficie: todas

Riesgo: bajo

Qué vas a aprender

- Escribir una petición útil con cuatro elementos.
- Expresar límites que previenen trabajo no deseado.
- Mejorar un resultado mediante mensajes de seguimiento.

Habla como con una persona colaboradora

No existe una fórmula obligatoria ni palabras mágicas. Una petición corta puede funcionar. Cuando el resultado importa, añade cuatro piezas:

1. **Objetivo:** qué resultado necesitas.
2. **Contexto:** qué archivos, fuentes o situación cambian la respuesta.
3. **Salida:** forma, longitud, audiencia y lugar de entrega.
4. **Límites:** qué debe conservar, evitar o consultar antes de hacer.

Una regla mnemotécnica es **OCSL: Objetivo, Contexto, Salida, Límites**.

De una petición vaga a un encargo comprobable

Petición vaga:

Organiza mis cosas.

Encargo útil:

OBJETIVO: convertir entrada/tareas.txt en un plan semanal.
CONTEXTO: las tareas no tienen fechas; no inventes plazos.
SALIDA: prepara primero una propuesta en el chat con tres categorías y una tabla de lunes a viernes.
LÍMITES: no cambies archivos todavía. Marca como "sin fecha" lo que no puedas colocar y pregúntame antes de crear el resultado final.

El segundo encargo no controla cada paso interno. Define las decisiones que podrían causar un resultado incorrecto.

Práctica guiada

Utiliza entrada/tareas.txt y envía el encargo anterior. Revisa la propuesta. Después responde con una corrección concreta:

Conserva las categorías. Mueve las tareas administrativas al martes, no asignes más de tres tareas a un día y añade una sección final con elementos que necesitan mi decisión. Todavía no crees archivos.

Cuando estés conforme:

Crea `salida/plan-semanal.md` con la versión aprobada. No modifiques los archivos de entrada. Después vuelve a abrir el archivo creado y comprueba que contiene las tres categorías, los cinco días y la sección de decisiones pendientes.

Resultado esperado

Un archivo nuevo dentro de `salida`, mientras `entrada` permanece intacta. El resumen final debe indicar qué comprobó y qué no pudo decidir.

Comprobación manual

- Abre `salida/plan-semanal.md`.
- Cuenta los cinco días.
- Confirma que ningún día tiene más de tres tareas.
- Busca la sección de decisiones pendientes.
- Abre `entrada/tareas.txt` y confirma que no cambió.

Recuperación

Si el resultado no sirve, no necesitas borrar la conversación. Especifica el cambio y pide guardar una segunda versión, por ejemplo `plan-semanal-v2.md`. Elimina la anterior únicamente después de comparar ambas.

Error frecuente

Describir veinte reglas que no cambian el resultado. Demasiadas instrucciones pueden ocultar las restricciones verdaderamente importantes. Prioriza audiencia, fuentes, formato y acciones que requieren permiso.

Mini ejercicio

Convierte una petición tuya en OCSL. Después elimina una instrucción que no cambie el resultado.

Evidencia que debes guardar

- Petición inicial.
- Mensaje de corrección.
- Lista de comprobación final.

Ejemplo resuelto: de «ordénamelo» a un encargo comprobable

Marta empieza con este prompt:

Ordéname estos clientes y déjalo bonito.

El problema no es que sea corto, sino que oculta decisiones: ¿qué archivos?, ¿qué significa ordenar?, ¿se pueden cambiar originales?, ¿cuál es el resultado?, ¿cómo se comprueba?

Aplicando OCSL -objetivo, contexto, salida y límites- lo transforma:

Objetivo: crear un resumen de seguimiento de clientes.
Contexto: usa únicamente los CSV de la carpeta entrada. La columna `ultima_contacto` está en formato AAAA-MM-DD.
Salida: crea `salida/seguimiento.csv` ordenado por fecha ascendente y `salida/resumen.md` con total de clientes, pendientes y filas incompletas.
Límites: no modifiques entrada, no inventes datos y no envíes mensajes.

Antes de escribir, muestra las columnas detectadas y tres filas de vista previa.
Al terminar, comprueba que el número de filas coincide y enumera los archivos creados.

Ahora «correcto» es observable: los originales continúan intactos, el recuento coincide, los datos faltantes se señalan y solo aparecen dos archivos en salida.

Una técnica útil: sustituye adjetivos vagos por pruebas. «Bonito» puede convertirse en «títulos claros, una tabla legible y fechas con el mismo formato». «Profesional» puede convertirse en «sin errores ortográficos, tono formal y llamadas a la acción visibles».

Criterio de terminado: otra persona puede leer tu prompt y predecir qué se creará, qué no se tocará y cómo se validará.

Cambia una variable

- **Sol:** cambia «hazme apuntes buenos» por una guía con nivel, extensión, conceptos obligatorios y cinco preguntas.
- **Leo:** cambia «investiga mucho» por preguntas, periodo, tipos de fuente y tabla de afirmaciones con enlaces.
- **Dani:** cambia «arregla el código» por error reproducible, alcance, prueba y archivos excluidos.

Comprueba que lo has entendido

1. ¿Todos los prompts largos son mejores?
2. Convierte «haz una web moderna» en dos criterios observables.

Solución comentada

No. La longitud no sustituye la claridad. Dos criterios posibles son: «se adapta a móvil sin desplazamiento horizontal» y «contiene navegación, descripción del servicio y un contacto visible». Hay muchas respuestas válidas si pueden comprobarse.

Fuentes oficiales

- [Prompting](#)
- [Best practices](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 6. Planifica, corrige el rumbo y cierra con evidencia

Tiempo: 40 minutos

Superficie: aplicación, CLI o IDE

Riesgo: bajo

Qué vas a aprender

- Pedir un plan antes de una tarea ambigua.
- Distinguir entre corregir el trabajo actual y dejar una petición en cola.
- Definir cuándo una tarea está realmente terminada.

Plan primero cuando decidir cuesta más que ejecutar

Pide un plan cuando el trabajo tiene varias salidas posibles, afecta a muchos archivos o contiene decisiones que no quieres delegar. Puedes utilizar Plan mode si aparece en tu superficie, escribir `/plan` donde esté disponible o pedirlo en lenguaje natural:

No actúes todavía. Inspecciona el material, hazme las preguntas que cambien el resultado y propón un plan de máximo cinco pasos. Para cada paso indica qué archivo podría cambiar y cómo comprobaríamos que está bien.

Un buen plan no es una lista genérica de verbos. Debe revelar decisiones, archivos, riesgos y comprobaciones.

Corregir mientras trabaja

Si Codex está trabajando y descubres un error, envía un mensaje para **steer** o dirigir el turno actual:

Detente antes de crear archivos. Has supuesto que las tareas tienen prioridad; el archivo no contiene esa información. Marca la prioridad como pendiente y continúa con el resto del plan.

Si tu mensaje pertenece al paso siguiente, colócalo en **queue** o cola. En la aplicación puedes elegir el comportamiento en Settings > General > Follow-up behavior. En CLI, Enter dirige el turno activo y Tab coloca el mensaje para el siguiente turno.

«Terminado» necesita una definición

Una respuesta elocuente no demuestra que el resultado esté listo. Antes de autorizar el trabajo, define condiciones observables:

La tarea estará terminada cuando:

- exista salida/resumen.md;
- contenga exactamente las tres secciones acordadas;
- cada afirmación procedente de notas.txt cite el encabezado de origen;
- entrada permanezca sin cambios;
- indiques cualquier comprobación que no pudiste realizar.

Práctica: plan y cierre

Pide a Codex que combine ideas.txt y notas.txt en un resumen. Primero solicita plan y preguntas. Responde solo a las preguntas necesarias. Autoriza la creación cuando el plan sea verificable.

Comprobación manual

Utiliza una tabla sencilla:

Condición	Sí	No	No comprobado
Existe el archivo acordado			
Mantiene las secciones			
Conserva los originales			
Declara sus límites			

No conviertas «No comprobado» en «Sí» por intuición. Esa columna es información útil.

Recuperación

Si Codex se desvía, interrumpe la tarea y vuelve al último punto aprobado. No amplíes permisos para intentar que una tarea mal definida termine más rápido.

Error frecuente

Cambiar el objetivo a mitad del trabajo sin decir qué queda cancelado. Escribe: «Sustituye el objetivo anterior por este; conserva estos archivos; descarta estos pasos».

Mini ejercicio

Pide dos planes alternativos para la misma tarea: uno rápido y otro con mayor verificación. Elige justificando coste y riesgo.

Evidencia que debes guardar

- Plan aprobado.
- Mensajes que corrigieron el rumbo.
- Tabla de condiciones finales.
- Límites declarados por Codex.

Ejemplo resuelto: corregir el rumbo sin empezar de cero

Leo encarga una comparación de cinco fuentes. Codex propone este plan: buscar fuentes, resumirlas, redactar una conclusión y crear el informe. Durante la búsqueda, Leo observa que dos resultados son notas comerciales y no documentos primarios.

En lugar de esperar al final escribe una corrección inmediata:

Corrige el criterio de fuentes antes de continuar: prioriza documentación oficial y estudios originales. Conserva las fuentes comerciales en una sección aparte llamada «perspectiva del proveedor» y no las utilices para sostener la conclusión principal. Muéstrame la lista revisada antes de redactar.

Esto es **steer**: modifica el trabajo activo. Después añade a la cola:

Cuando termines la verificación, crea también una tabla con afirmación, fuente, fecha y nivel de confianza.

Al cerrar, no pregunta «¿todo bien?». Pide un parte de entrega:

Resume qué hiciste, qué archivos cambiaste, qué comprobaste realmente, qué no pudiste comprobar y qué decisión necesita revisión humana.

La lección es doble: un plan no es una promesa inmutable y una respuesta final no es evidencia por sí sola.

Señal de alarma: el plan se aleja del objetivo, pero dejas que avance porque «ya ha empezado».

Criterio de terminado: puedes reconstruir el recorrido desde el encargo, las correcciones y la evidencia final.

Cambia una variable

Pide a Codex un plan deliberadamente corto de tres pasos. Durante el segundo, introduce una restricción nueva compatible, como excluir un archivo de salida. Observa si actualiza el plan y confirma el efecto.

Comprueba que lo has entendido

1. ¿Qué diferencia hay entre corregir el trabajo activo y añadir una tarea posterior?
2. ¿«Las pruebas pasan» es evidencia suficiente?

Solución comentada

Corregir el trabajo activo cambia el rumbo actual; poner algo en cola lo ejecuta después. «Las pruebas pasan» solo es útil si sabes qué prueba se ejecutó, con qué resultado y si realmente cubre el cambio relevante.

Fuentes oficiales

- [Best practices](#)
- [Prompting](#)
- [ChatGPT desktop app settings](#)

Revisado contra la documentación: 14 de julio de 2026.

Módulo 3. Trabajo cotidiano sin programar

Lección 7. Organiza archivos sin perder los originales

Tiempo: 45 minutos

Superficie: Codex local

Plataformas: macOS, Windows y Linux

Qué vas a aprender

- Inventariar archivos antes de cambiarlos.
- Separar propuesta y ejecución.
- Detectar nombres duplicados y conservar una copia recuperable.

Organizar no empieza moviendo

Una tarea como «ordena esta carpeta» es peligrosa porque oculta decisiones: qué criterio utilizar, qué hacer con duplicados, si se pueden borrar archivos y cómo recuperar la estructura anterior.

Utiliza siempre tres fases:

1. **Inventario:** qué existe y qué no se puede interpretar.
2. **Vista previa:** tabla exacta de origen y destino, sin cambios.
3. **Ejecución:** copiar primero; borrar o reemplazar solo después de revisar.

Prepara material de entrenamiento

Dentro de Aprender-Codex, crea archivos-desordenados y copia allí seis archivos desechables. Puedes duplicar los textos del curso y darles nombres como:

- notas reunion.txt
- NOTAS-reunion-copia.txt
- idea final.txt
- idea final 2.txt
- tareas julio.txt
- sin nombre.txt

Crea también archivos-organizados. No utilices todavía fotos ni documentos personales.

Fase 1 - inventario

Analiza únicamente archivos-desordenados. No cambies nada. Devuelve una tabla con:

- nombre actual;
- tipo de archivo;
- tema probable;
- posibles duplicados;
- información que no puedes determinar.

No deduzcas que dos archivos son duplicados solo porque sus nombres se parecen.

Fase 2 - vista previa

Propón una organización dentro de archivos-organizados con subcarpetas por tema.

Usa nombres en minúsculas, guiones y sin fechas inventadas. Devuelve una tabla origen → destino. Señala colisiones, pero no copies, muevas, renombres ni borres.

Revisa cada fila. Si dos archivos terminarían con el mismo nombre, exige nombres distintos o detén la ejecu-

ción.

Fase 3 - copia controlada

Aplica únicamente la tabla que acabo de aprobar. Copia los archivos a `archivos-organizados`; no muevas ni borres los originales. Después compara el número de archivos de origen y destino, informa de cualquier error y guarda en `salida/mapa-organizacion.md` la tabla realmente aplicada.

Resultado esperado

Los originales siguen en `archivos-desordenados`, las copias están clasificadas en `archivos-organizados` y existe un mapa de lo realizado. El número de archivos debe coincidir salvo que hayas aprobado una excepción.

Comprobación manual

- Cuenta los archivos de origen y las copias.
- Abre al azar dos copias y compáralas con sus originales.
- Busca nombres repetidos dentro de cada carpeta de destino.
- Comprueba que el mapa coincide con las ubicaciones reales.

Recuperación

Como los originales no se movieron, puedes eliminar `archivos-organizados` y repetir la práctica. No autorices el borrado de `archivos-desordenados` durante el curso.

Error frecuente

Pedir que elimine duplicados sin definir duplicado. Dos archivos pueden compartir nombre y contener información distinta. Primero compara contenido, tamaño o una regla explícita; después decide tú.

Mini ejercicio

Añade un séptimo archivo cuyo destino chocaría con otro. Pide una nueva vista previa y comprueba que Codex detiene o resuelve la colisión sin sobrescribir.

Evidencia que debes guardar

- Inventario inicial.
- Tabla aprobada.
- Mapa aplicado.
- Conteo antes y después.

Ejemplo resuelto: organizar sin destruir el significado

Leo tiene doce archivos con nombres como `doc1.pdf`, `version-final.docx` y `captura 3.png`. Pedir «organiza» podría provocar una clasificación arbitraria. Empieza por inventario y propuesta:

No cambies nada todavía. Crea una tabla con nombre actual, tipo, fecha disponible, tema probable y nombre propuesto. Si el tema no se puede deducir con seguridad, marca REVISAR. Propón una estructura de carpetas con un máximo de dos niveles. Detecta colisiones antes de renombrar.

La propuesta incluye fuentes, borradores y recursos-visuales. Dos documentos recibirían el mismo nombre. Codex no debe sobrescribir uno: propone sufijos descriptivos y solicita decisión.

Leo aprueba solo cuatro cambios y pide que el resto permanezca intacto. Codex crea primero una copia en organizado–vista–previa y entrega un archivo mapa–de–cambios.md con nombre anterior y nuevo. Leo abre una muestra, comprueba que las extensiones se conservan y compara el total.

Regla práctica: una buena organización conserva cuatro cosas: contenido, cantidad, relación entre archivos y posibilidad de localizar el original.

Señal de alarma: aparecen nombres precisos que no pueden justificarse por el contenido, o se propone borrar «duplicados» sin comparar su identidad.

Criterio de terminado: el recuento cuadra, no hay sobrescrituras, el mapa permite deshacer manualmente y una muestra se abre correctamente.

Cambia una variable

- **Sol:** clasifica por asignatura y tipo, pero no por una fecha inexistente.
- **Marta:** separa documentos de clientes sin incluir datos personales en nombres visibles.
- **Dani:** excluye dependencias generadas y archivos ocultos antes de reorganizar un proyecto.

Comprueba que lo has entendido

1. ¿Dos archivos con el mismo tamaño son necesariamente duplicados?
2. ¿Por qué conviene un mapa de nombres antes/después?

Solución comentada

No. Pueden contener datos distintos; la deduplicación exige una comparación fiable. El mapa documenta el cambio, permite localizar archivos y facilita una reversión si la clasificación no funciona.

Fuentes oficiales

- [Projects, chats, and tasks](#)
- [Work with files](#)
- [Sandbox](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 8. Crea documentos, datos, presentaciones, PDF e imágenes

Tiempo: 50 minutos

Superficie: ChatGPT Work y Codex

Plataformas: web y aplicación de escritorio

Qué vas a aprender

- Elegir Work para entregables visuales y Codex para preparar fuentes o trabajar en una carpeta.
- Pedir un archivo con estructura y criterios de revisión.
- Utilizar imágenes y capturas como contexto sin depender solo de ellas.

Divide el trabajo por tipo de resultado

La documentación oficial recomienda ChatGPT Work para documentos, presentaciones, hojas de cálculo, PDF, Sites e investigación. La aplicación de escritorio puede previsualizar esos archivos y permite señalar una parte concreta para pedir una revisión.

Codex es útil antes o después:

- prepara y limpia archivos fuente dentro de una carpeta;
- extrae información de un proyecto técnico;
- genera datos o recursos que formarán parte del entregable;
- actualiza archivos del proyecto después de aprobar el resultado.

No necesitas obligar a una sola tarea a hacerlo todo.

Práctica: de notas desordenadas a un informe revisable

1. En Codex, pide combinar `ideas.txt`, `notas.txt` y `plan-semanal.md` en `salida/fuentes-informe.md`.
2. Exige que cada apartado indique el archivo de origen y que no invente información.
3. Abre ChatGPT Work y adjunta `fuentes-informe.md`.
4. Pide un documento de dos páginas con título, resumen, tres apartados y próximos pasos.
5. Abre la previsualización y revisa cada página.

Encargo para Codex

Prepara `salida/fuentes-informe.md` a partir de `entrada/ideas.txt`, `entrada/notas.txt` y `salida/plan-semanal.md`. Conserva el significado, elimina repeticiones literales y añade después de cada apartado una línea "Fuente:" con la ruta correspondiente. Marca [INFORMACIÓN AUSENTE] en lugar de inventar datos. No modifiques los archivos de origen.

Encargo para Work

Crea un documento de dos páginas para una persona que no conoce el proyecto. Utiliza únicamente el archivo adjunto. Estructura: título, resumen de cinco líneas, tres apartados y próximos pasos. Conserva las marcas [INFORMACIÓN AUSENTE]. No conviertas una inferencia en un hecho. Antes de terminar, comprueba que cada apartado puede rastrearse a una fuente indicada en el material.

Hojas de cálculo y presentaciones

Cuando pidas una hoja, define hojas, columnas, fórmulas, unidades y comprobaciones. Para una presentación, define audiencia, número de diapositivas, mensaje por diapositiva y datos que deben citarse. Evita «hazlo bonito» como único criterio.

Imágenes y capturas

Una imagen puede mostrar un error, un diseño o un resultado visual. Explica qué zona importa y qué resultado esperas:

La captura muestra el informe abierto. Revisa únicamente si hay texto cortado, contraste insuficiente o elementos desalineados. No deduzcas datos que no se leen. Devuelve una lista de correcciones por página.

En CLI puedes adjuntar imágenes con `-i` o `--image`; en la aplicación y el IDE, sigue el control de adjuntos de la interfaz. No compartas capturas con notificaciones, nombres de cuenta o información de otras aplicaciones.

Comprobación manual

- Recorre cada página, diapositiva o pestaña.
- Verifica nombres, fechas, totales, citas y marcas de información ausente.
- Comprueba que el archivo se abre en una aplicación compatible.
- Revisa que títulos y tablas puedan entenderse sin depender únicamente del color.
- Pide una nueva versión con otro nombre; no sobrescribas la aprobada durante la revisión.

Recuperación

Conserva el archivo fuente y versiones numeradas del entregable. Si una revisión empeora otra parte, vuelve a la versión anterior y pide un cambio limitado a la página, diapositiva, hoja o fragmento afectado.

Error frecuente

Revisar solo la respuesta del chat. Un mensaje puede decir que el archivo está correcto mientras una página contiene texto cortado o una hoja utiliza una fórmula incorrecta. Abre siempre el entregable.

Mini ejercicio

Pide una versión de una página para dirección y otra de dos páginas para el equipo. Compara qué información se conserva y cuál se elimina.

Evidencia que debes guardar

- Archivo fuente consolidado.
- Primera y última versión del entregable.
- Lista de revisiones solicitadas.
- Comprobaciones que realizaste manualmente.

Ejemplo resuelto: separar contenido, cálculo y presentación

Sol necesita una guía didáctica con una tabla de resultados y seis diapositivas. En vez de pedirlo todo en un único salto, divide el trabajo:

1. En Codex prepara y limpia las fuentes de la carpeta, detecta datos ausentes y crea `contenido-validado.md` y `resultados-limpios.csv`.
2. Comprueba una muestra del CSV y recalcula dos valores con una calculadora o herramienta independien-

te.

3. Utiliza Work para convertir el contenido validado en documento y presentación revisables.
4. Exporta a PDF solo cuando títulos, tablas, cortes de página y créditos están aprobados.

Prompt de preparación:

Lee las notas y el CSV de esta carpeta. No diseñes todavía el documento.
Crea contenido-validado.md con objetivos, explicación, actividad y fuentes.
Crea resultados-limpios.csv conservando todas las filas; marca los datos ausentes, no los inventes. Incluye un informe de controles y recuentos.

Este orden evita ocultar errores detrás de una presentación atractiva. Un PDF visualmente impecable puede contener un cálculo equivocado; por eso la comprobación del dato precede al diseño.

Comprobación visual mínima: abre cada formato final, mira la primera y última página o diapositiva, verifica que no haya texto cortado, que las tablas sean legibles y que los enlaces o créditos aparezcan.

Criterio de terminado: los datos tienen controles, el contenido se puede rastrear a sus fuentes y cada formato final ha sido abierto y revisado.

Cambia una variable

Si el resultado incluye una imagen generada, añade una revisión de texto incrustado, manos/objetos deformes, atribución necesaria y adecuación al contexto. Si existe material de referencia, conserva su origen.

Comprueba que lo has entendido

1. ¿Por qué no basta con que un PDF «se vea profesional»?
2. ¿Qué debe ocurrir antes de convertir datos en gráficos o diapositivas?

Solución comentada

Porque la apariencia no demuestra exactitud, integridad ni trazabilidad. Antes de visualizar, se revisan estructura, valores ausentes, unidades, recuentos y una muestra de cálculos.

Fuentes oficiales

- [Work with files](#)
- [Image inputs](#)
- [Prompting](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 9. Investiga en la web y utiliza el navegador con criterio

Tiempo: 50 minutos

Superficie: web o aplicación de escritorio

Riesgo: medio

Qué vas a aprender

- Diferenciar búsqueda web de control del navegador.
- Pedir fuentes y comprobar que respaldan la afirmación.
- Protegerte frente a instrucciones maliciosas contenidas en páginas.

Búsqueda y navegador no son lo mismo

La **búsqueda web** encuentra información y devuelve resultados o citas. El **navegador** abre páginas, inspecciona su estado visible y, cuando lo autorizas, puede hacer clic o escribir.

Utiliza búsqueda cuando necesitas información. Utiliza Browser cuando necesitas observar o recorrer una página concreta. En CLI e IDE no está disponible el navegador integrado, aunque sí puede existir búsqueda. El navegador integrado de la aplicación utiliza un perfil separado de tu navegador habitual. La extensión de Chrome se utiliza cuando necesitas una pestaña o sesión de tu perfil de Chrome.

Las páginas son fuentes, no instrucciones para el agente

Una web puede contener texto como «ignora lo anterior», «pega aquí tu clave» o «descarga este programa». Ese contenido no recibe autoridad por aparecer en la página. Trátalo como información no confiable y mantén tus límites.

Práctica de investigación pública

Elige tres páginas públicas sobre un tema no sensible. Por ejemplo, documentación de una biblioteca, horarios de museos o condiciones de tres servicios gratuitos. Evita iniciar sesión, comprar, reservar o enviar formularios.

Compara estas tres páginas públicas: [URL 1], [URL 2] y [URL 3]. Necesito una tabla con el dato principal, fecha o vigencia visible, condiciones y enlace a la fuente exacta. Distingue hechos explícitos de inferencias. Si una página no es accesible o el dato no aparece, escribe "no verificado". No rellenes formularios, no descargues archivos y no realices ninguna acción externa.

Cómo comprobar una cita

1. Abre el enlace asociado a una afirmación importante.
2. Busca el dato concreto, no solo palabras relacionadas.
3. Comprueba fecha, moneda, unidad, región y excepciones.
4. Si la página es secundaria, busca la fuente primaria.
5. Marca como no verificado lo que no puedas localizar.

Una lista larga de enlaces no sustituye esta comprobación.

Cuando utilizar Browser

En la aplicación, instala o activa el plugin Browser si está disponible. Permite únicamente los sitios necesarios. Pide primero observación:

Usa Browser para abrir [URL]. Describe la navegación principal, los formularios visibles y cualquier aviso de privacidad. No hagas clic, no escribas, no aceptes cookies y no inicies sesión. Señala qué acciones requerirían mi confirmación.

Después de comprender la página, autoriza una acción concreta si es necesaria. Una autorización para visitar el sitio no equivale a permiso para enviar información, cambiar ajustes o borrar datos.

Limitaciones importantes

- Algunos sitios bloquean automatización o muestran CAPTCHA.
- El navegador integrado no automatiza cargas de archivos.
- La disponibilidad depende de plan, organización y despliegue.
- Un sitio permitido no se convierte en confiable.

Recuperación

Si el navegador entra en una ruta incorrecta, detén la tarea. Cierra la pestaña o vuelve a una página conocida. Si se envió información o cambió una cuenta, revisa inmediatamente el historial o ajustes del servicio y cambia credenciales cuando corresponda.

Error frecuente

Pedir «busca en internet» y aceptar el primer resumen. Define actualidad, jurisdicción, fuentes preferidas y datos que deben quedar como no verificados.

Mini ejercicio

Elige una afirmación de la tabla y compruébala manualmente. Escribe qué parte de la fuente la respalda y qué excepción podría cambiarla.

Evidencia que debes guardar

- Consulta utilizada.
- Tabla con enlaces directos.
- Una afirmación comprobada manualmente.
- Lista de datos no verificados.

Ejemplo resuelto: una investigación que se puede auditar

Leo quiere comparar tres herramientas para una asociación. Un encargo débil sería «busca cuál es la mejor». Lo convierte en un protocolo:

Compara las tres herramientas indicadas para una asociación de 20 personas. Responde estas preguntas: precio actual, límites del plan, exportación de datos, accesibilidad y soporte. Prioriza páginas oficiales y documentación primaria. Registra URL, título, fecha de publicación o actualización si aparece y fecha de consulta. Separa hechos, inferencias y datos no confirmados. No recomiendes hasta mostrar la tabla de evidencia.

Codex encuentra un precio en una reseña reciente y otro distinto en la página oficial. Leo no promedia los valores: conserva el oficial como evidencia principal, registra la discrepancia y anota que el precio puede depender

de región o impuestos.

La tabla utiliza cinco columnas: afirmación, evidencia, fuente, vigencia y confianza. La conclusión solo incluye afirmaciones que aparecen en la tabla.

Prueba de honestidad: elige una frase importante del resumen y recorre el camino inverso hasta la página que la respalda. Si no puedes hacerlo en menos de un minuto, falta trazabilidad.

Señal de alarma: una página secundaria cita a otra que cita a una tercera, pero el informe presenta el dato como comprobado en la fuente original.

Criterio de terminado: cada afirmación decisiva tiene una fuente accesible, se distingue la fecha del hecho de la fecha del artículo y las incertidumbres son visibles.

Cambia una variable

- Para una noticia, compara la fecha de publicación con la fecha del acontecimiento.
- Para documentación técnica, registra la versión a la que se refiere.
- Para una recomendación costosa, añade criterios personales y no delegues la decisión final.

Comprueba que lo has entendido

1. ¿Una fuente reciente siempre es mejor que una fuente primaria más antigua?
2. ¿Qué diferencia hay entre un hecho y una inferencia?

Solución comentada

No automáticamente: depende de la pregunta, la vigencia y la autoridad. Un hecho está directamente respaldado por la evidencia; una inferencia es una conclusión razonable derivada de uno o más hechos y debe etiquetarse como tal.

Fuentes oficiales

- [Web search](#)
- [Browser](#)
- [Prompting](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 10. Usa Computer Use sin entregar el control de tu equipo

Tiempo: 45 minutos

Superficie: aplicación de escritorio

Plataformas: macOS y Windows

Riesgo: medio

Qué vas a aprender

- Cuándo una tarea necesita ver y operar una interfaz gráfica.
- Qué permisos concede Computer Use.
- Cómo preparar una práctica limitada y reversible.

Qué hace Computer Use

Computer Use permite a ChatGPT ver y operar aplicaciones gráficas autorizadas. Puede hacer clic, escribir, abrir menús e inspeccionar una interfaz. Es útil cuando no existe una integración estructurada y el resultado depende de lo que aparece en pantalla.

No es el primer recurso para todo. Si existe un plugin, una API o un archivo que proporciona los datos directamente, esa vía suele ser más repetible y fácil de auditar.

Diferencias de plataforma

macOS: requiere permisos de Screen Recording para ver y Accessibility para interactuar. Puedes retirar esos permisos desde Privacy & Security. Algunas tareas pueden ejecutarse en segundo plano según la configuración.

Windows: trabaja sobre el escritorio activo y puede mover el puntero y escribir en primer plano. Mantén visible la aplicación objetivo y no utilices al mismo tiempo esa sesión de Windows.

Computer Use no puede aprobar por ti los permisos de seguridad del sistema ni autenticarse como administrador. Tampoco debe automatizar ChatGPT o aplicaciones de terminal.

Preparación segura

1. Cierra correo, gestor de contraseñas, banca, mensajería y documentos sensibles.
2. Abre una aplicación de texto con un documento nuevo y vacío.
3. Instala o activa Computer Use desde Plugins si aparece en tu cuenta.
4. Autoriza únicamente la aplicación de texto.
5. No selecciones «Always allow» durante la primera práctica.

Práctica

Usa Computer Use únicamente con [TextEdit/Bloc de notas]. En el documento vacío, escribe el título "Práctica de Computer Use" y debajo tres viñetas sobre lo que puedes ver en esa ventana. No abras otros archivos, no cambies ajustes, no guardes, no cierres la aplicación y no uses el portapapeles. Detente y muéstrame el resultado antes de cualquier otra acción.

Revisa el texto. Si quieres conservarlo, guárdalo tú manualmente dentro de Aprender-Codex/salida. El curso no necesita que el agente pulse Guardar.

Comprobación manual

- Confirma que solo se utilizó la aplicación autorizada.
- Revisa el texto antes de guardar.
- Comprueba que no hay ventanas sensibles visibles en la captura o el escritorio.
- Abre Settings > Computer Use y revisa qué aplicaciones tienen permiso persistente.

Recuperación

Puedes detener la tarea o tomar el control en cualquier momento. Si interactúa con la ventana incorrecta, cancela inmediatamente. Retira la aplicación de Always-allowed apps y, en macOS, revoca Screen Recording o Accessibility si ya no utilizarás la función.

Acciones que requieren presencia humana

Mantente presente para cuentas, credenciales, privacidad, pagos, permisos, redes, publicaciones y cualquier acción que afecte a terceros. Una tarea que llega a un botón final de enviar, comprar, borrar o publicar debe detenerse antes de pulsarlo salvo autorización explícita y actual.

Error frecuente

Autorizar una aplicación cuando hay otras ventanas sensibles visibles. Computer Use procesa lo que aparece en pantalla. Prepara el entorno antes de iniciar.

Mini ejercicio alternativo

Si Computer Use no está disponible, describe con una captura recortada una interfaz de práctica y pide una lista de pasos sin ejecutar acciones.

Evidencia que debes guardar

- Aplicación autorizada.
- Resultado visible antes de guardar.
- Revisión de permisos persistentes.
- Cualquier limitación de tu plataforma.

Ejemplo resuelto: utilizar la interfaz sin ceder una decisión

Marta quiere descargar una factura de un portal. La tarea contiene navegación, una posible sesión iniciada y datos privados. La divide en puntos de control:

Abre el portal y navega hasta la sección de facturas.
No cambies datos de cuenta, no aceptes condiciones, no envíes formularios y no descargues todavía. Detente cuando veas la factura de junio y descríbeme la página, el importe y el botón que usarías. Espera mi confirmación.

Marta verifica visualmente dominio, periodo e importe. Solo entonces autoriza la descarga a una carpeta concreta. Después abre el PDF y confirma que corresponde al mes correcto.

No sería apropiado ampliar el encargo a «y paga cualquier importe pendiente». El pago es una acción externa con consecuencias y necesita una autorización específica, datos correctos y revisión en el momento de confirmar.

Tres niveles de acción: observar una pantalla, preparar una acción y ejecutarla. Mantenerlos separados evita que una instrucción inocente termine en un envío o compra.

Señal de alarma: aparece una pantalla inesperada de contraseña, segundo factor, consentimiento, compra, publicación o eliminación. Detén la automatización y revisa personalmente.

Criterio de terminado: la acción autorizada es exactamente la realizada, el archivo está en la ubicación acordada y no hubo cambios colaterales en la cuenta.

Cambia una variable

- **Sol:** puede preparar el borrador de un formulario, pero revisa destinatario y respuestas antes de enviarlo.
- **Leo:** puede recopilar títulos visibles, pero no sortear controles de acceso.
- **Dani:** para una consola web, separa inspección, cambio de configuración y publicación.

Comprueba que lo has entendido

1. ¿Por qué conviene detenerse justo antes de pulsar un botón irreversible?
2. ¿Una sesión ya iniciada convierte todas las acciones en autorizadas?

Solución comentada

La pausa permite comprobar contexto, destinatario, importe y efecto antes de una consecuencia real. Una sesión iniciada solo aporta acceso técnico; no concede autorización general para actuar.

Fuentes oficiales

- [Computer Use](#)
- [Browser](#)
- [Permissions](#)

Revisado contra la documentación: 14 de julio de 2026.

Prueba pendiente antes de publicación: permisos y nombres de menús actuales en macOS y Windows.

Módulo 4. Construir sin saber programar

Lección 11. Construye tu primera web local

Tiempo: 75 minutos

Superficie: Codex local; navegador para revisar

Plataformas: macOS, Windows y Linux

Qué vas a aprender

- Convertir una idea en un proyecto técnico pequeño sin escribir código manualmente.
- Revisar el resultado como usuario.
- Separar «funciona en mi equipo» de «está publicado».

El proyecto

Construirás una página de presentación que funciona localmente. No tendrá formularios, analítica, pagos, cuentas, cookies ni dependencias externas. Continuará dentro de tu ordenador: crear una web local no significa publicarla.

Dentro de Aprender-Codex, crea mi-primer-web.

Fase 1 - define el contenido

Decide:

- nombre del proyecto;
- audiencia;
- frase principal;
- tres servicios o ideas;
- forma de contacto ficticia;
- dos colores que no sean obligatorios para comprender el contenido.

No utilices dirección, teléfono ni correo reales en la práctica.

Fase 2 - encargo

Construye dentro de mi-primer-web una página local de una sola pantalla para [proyecto]. Audiencia: [audiencia]. Contenido obligatorio: [contenido].

Requisitos:

- usa HTML y CSS sencillos, sin dependencias, CDN, analítica ni red;
- crea index.html y styles.css;
- diseño adaptable a móvil y escritorio;
- HTML semántico, foco visible, contraste legible y enlace para saltar al contenido;
- no uses texto de relleno ni inventes testimonios;
- el contacto será contacto@example.com y debe indicarse que es ficticio;
- no publiques ni abras puertos de red.

Primero muestra el plan y los archivos que crearás. Espera mi aprobación antes de escribir. Después valida la estructura y explícame cómo abrirla localmente.

Fase 3 - revisión

Abre index.html en un navegador. En la aplicación de escritorio puedes utilizar el navegador integrado; en Linux puedes abrir el archivo con tu navegador habitual.

Comprueba:

- el título coincide con el proyecto;
- las tres ideas están presentes;
- el correo está marcado como ficticio;
- la página se entiende con imágenes desactivadas;
- se puede recorrer con Tab y el foco es visible;
- al estrechar la ventana no aparece desplazamiento horizontal;
- no hay enlaces que envíen datos o publiquen contenido.

Feedback preciso

Evita «hazla más moderna». Describe el estado observado y el esperado:

En una ventana estrecha, el segundo bloque se sale por la derecha. Mantén el contenido y los colores; corrige únicamente el ancho y el ajuste de texto. Después vuelve a comprobar móvil y escritorio.

Resultado esperado

Dos archivos locales, una página legible y una explicación de las comprobaciones. El agente no debe desplegar, comprar dominio ni añadir servicios externos.

Recuperación

Antes de una revisión amplia, copia la carpeta como `mi-primer-web-aprobada`. Si la nueva versión empeora, restaura esos dos archivos. En la lección 15 aprenderás una forma más potente de guardar versiones con Git.

Error frecuente

Confundir construir con publicar. Un despliegue puede crear una URL pública, costes o datos accesibles. Es una tarea distinta que requiere revisar contenido, privacidad y proveedor.

Mini ejercicio

Añade un bloque de preguntas frecuentes con dos preguntas. Exige que no cambie ninguna sección aprobada y verifica el orden de navegación con teclado.

Evidencia que debes guardar

- Encargo inicial.
- Captura de móvil y escritorio.
- Lista de comprobaciones.
- Copia de la versión aprobada.

Ejemplo resuelto: una web pequeña con una definición de terminado

Marta quiere una página para presentar un servicio de asesoría. Antes de pedir código prepara un encargo de una página:

Construye una web local de una sola página para un servicio de asesoría. Público: pequeños comercios. Tono: claro, cercano y sin promesas exageradas. Secciones obligatorias: cabecera, problema, tres servicios, proceso en tres pasos, preguntas frecuentes y contacto. Usa contenido provisional claramente marcado; no inventes testimonios, clientes ni certificaciones. Debe funcionar en móvil y escritorio, tener navegación por teclado, contraste legible y ningún recurso externo necesario para abrirla.

Primero propón estructura y archivos. Después implementa. Al terminar, ejecuta las comprobaciones disponibles y explicame cómo abrirla localmente.

Codex propone `index.html`, `styles.css` y, solo si hace falta, `script.js`. Marta revisa la estructura antes de autorizar la implementación. Al terminar no se limita a mirar una captura: abre la página, reduce el ancho de la ventana, utiliza Tab para recorrer enlaces, prueba el menú y confirma que los textos provisionales se distinguen.

Después solicita una corrección concreta: «En móvil, el título ocupa demasiado. Reduce su tamaño sin cambiar el texto y comprueba que no aparezca desplazamiento horizontal». Una modificación pequeña y verificable enseña más que pedir «hazla más moderna».

Lista de aceptación: abre sin conexión, no muestra errores visibles, contiene todas las secciones, se puede recorrer con teclado, no inventa datos y explica cómo sustituir el contenido provisional.

Señal de alarma: la página incluye formularios que parecen enviar datos, analítica, recursos remotos o afirmaciones comerciales que no fueron autorizados.

Criterio de terminado: otra persona puede abrir la web siguiendo las instrucciones y completar la lista de aceptación.

Cambia una variable

- **Sol:** crea una página de recursos para una asignatura con lectura fácil.
- **Leo:** crea una ficha de proyecto con fuentes y licencia de imágenes.
- **Dani:** añade una prueba automática simple solo después de que los criterios sean claros.

Comprueba que lo has entendido

1. ¿Por qué conviene definir contenido y criterios antes de elegir animaciones o colores?
2. ¿Una captura demuestra que la web funciona en móvil y con teclado?

Solución comentada

La estructura y los criterios determinan si la web cumple su propósito; el estilo no repara un contenido incompleto. Una captura solo muestra un estado visual: la adaptación, navegación y comportamiento deben probarse.

Fuentes oficiales

- [Prompting](#)
- [Browser](#)
- [Work with files](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 12. Automatiza una tarea repetitiva con vista previa

Tiempo: 75 minutos

Superficie: Codex local

Plataformas: macOS, Windows y Linux

Qué vas a aprender

- Convertir un resultado manual en un proceso repetible.
- Utilizar las herramientas ya instaladas sin descargar software innecesario.
- Probar una automatización con datos conocidos y salidas separadas.

Automatizar no significa perder el control

Una automatización segura tiene una entrada, una salida, una regla conocida, un modo de prueba y una señal clara de fallo. No debe sobrescribir el origen ni esconder errores.

Dentro de Aprender-Codex/entrada, crea gastos-practica.csv:

```
fecha,categoria,concepto,importe
2026-07-01,material,cuadernos,24.50
2026-07-02,transporte,bus,18.00
2026-07-03,formacion,curso,42.25
2026-07-04,material,boligrafos,15.25
```

Son datos ficticios. El total conocido es 100.00; material suma 39.75, transporte 18.00 y formación 42.25.

Fase 1 - inspecciona el entorno

No instales nada y no cambies archivos. Comprueba qué herramienta ya disponible en este equipo serviría para leer un CSV y crear un resumen reproducible. Elige la opción más sencilla y multiplataforma posible. Si no hay una opción adecuada, detente y explica alternativas. No uses red.

Codex puede detectar Python, Node.js, PowerShell u otra herramienta. La elección puede variar; lo importante es que explique qué encontró. No autorices una instalación solo para completar la práctica.

Fase 2 - diseña antes de ejecutar

Propón una automatización que lea entrada/gastos-practica.csv y escriba archivos nuevos en salida/resumen-gastos. Debe producir un resumen por categoría, total general, número de filas válidas y lista de errores. No debe modificar el CSV. Incluye modo de prueba o vista previa, códigos de error y una instrucción breve para repetirla. Muéstrame los archivos y el comando previsto antes de crearlos.

Fase 3 - prueba con una copia

Autoriza la creación únicamente dentro de una carpeta nueva. Pide ejecutar primero la vista previa y comparar con los totales conocidos. Después autoriza la salida final.

Ejecuta primero el modo de vista previa. Compara el total y las categorías con estos valores esperados: total 100.00; material 39.75; transporte 18.00; formación 42.25. Si no coinciden, no escribas la salida final. Si coinciden, crea el resumen y vuelve a leerlo para verificarlo.

Prueba de error

Copia el CSV como gastos-error.csv y cambia un importe por NO_VALIDO. Ejecuta la automatización sobre esa copia. Debe informar del error y evitar presentar un total incompleto como correcto.

Comprobación manual

- El CSV original sigue igual.
- La vista previa coincide con 100.00.
- El resumen identifica cuatro filas válidas.
- La prueba defectuosa produce un error visible.
- Las instrucciones permiten repetir el proceso sin recordar la conversación.

Recuperación

Elimina solo la carpeta de salida y vuelve a ejecutar. Como la entrada está intacta, el proceso es recuperable. Si la automatización modifica el origen, no cumple el diseño y debe corregirse antes de reutilizarla.

Error frecuente

Probar únicamente el caso correcto. Una automatización puede parecer fiable hasta que encuentra una celda vacía, un nombre duplicado o un formato inesperado. Incluye al menos una prueba de fallo.

Mini ejercicio

Añade una quinta fila válida y calcula manualmente el nuevo total. Ejecuta de nuevo y comprueba que no necesitas modificar el programa para aceptar otra fila.

Evidencia que debes guardar

- Herramienta elegida y motivo.
- Plan previo.
- Comando o acción reproducible.
- Salida correcta y salida de error.
- Confirmación de que el origen no cambió.

Ejemplo resuelto: automatizar un CSV con una fila problemática

Marta recibe cada viernes un CSV con cliente, email, ultima_contacto y estado. Quiere un resumen de seguimientos. Una fila contiene una fecha inválida y otra no tiene correo. En vez de ordenar directamente, establece tres fases.

Fase 1 - inspección. Codex informa de columnas, codificación, número de filas, valores vacíos, formatos detectados y anomalías. No escribe resultados.

Fase 2 - vista previa. Procesa cinco filas y muestra cómo interpretará fechas, estados y ausencias. Marta confirma que una fecha ambigua no debe corregirse por intuición: se marca REVISAR.

Fase 3 - ejecución. Crea salidas nuevas y un informe de control.

Usa entrada/clientes.csv. No modifiques el original.
 Antes de ejecutar, valida cabeceras y muestra cinco filas transformadas.
 Después crea salida/seguimiento.csv con todas las filas, ordenadas por ultima_contacto válida; coloca al final las fechas inválidas y marca REVISAR.
 Crea salida/resumen.md con recuentos, incidencias y reglas aplicadas.
 Comprueba que filas_de_entrada = filas_de_salida y que no se enviaron correos.

Para demostrar que la automatización es reutilizable, Marta guarda una copia del encargo y lo repite con un segundo archivo de prueba. Si el nuevo CSV cambia el nombre de una columna, la tarea debe detenerse y explicar la incompatibilidad; no adaptar silenciosamente la regla.

Control de reconciliación: total de entrada, total procesado, total correcto y total a revisar deben ser matemáticamente compatibles.

Señal de alarma: el sistema «rellena» correos, fechas o estados que no estaban en la fuente.

Criterio de terminado: los originales siguen iguales, los recuentos cuadran, las excepciones son visibles y una segunda ejecución no duplica ni mezcla salidas.

Cambia una variable

Introduce deliberadamente una cabecera mal escrita. El comportamiento deseado es una parada explicada, no un resultado aparentemente válido.

Comprueba que lo has entendido

1. ¿Por qué una vista previa de cinco filas no sustituye el control final de todas las filas?
2. ¿Qué propiedad debe tener una automatización que se ejecutará cada semana?

Solución comentada

La muestra ayuda a validar la regla, pero puede no contener las excepciones del conjunto completo. Una automatización recurrente debe ser reproducible, detectar cambios de formato, separar entradas y salidas y producir controles revisables.

Fuentes oficiales

- [Sandbox](#)
- [Agent approvals & security](#)
- [Integrated terminal](#)

Revisado contra la documentación: 14 de julio de 2026.

Prueba pendiente antes de publicación: ejecutar la práctica con herramientas disponibles por defecto o habitualmente disponibles en macOS, Windows y Linux y documentar las variantes reales.

Módulo 5. Organización, control y recuperación

Lección 13. Organiza proyectos, tareas y entornos

Tiempo: 45 minutos

Superficie: aplicación, web, CLI o IDE

Plataformas: macOS, Windows y Linux

Qué vas a aprender

- Diferenciar proyecto, tarea, conversación y carpeta.
- Separar resultados distintos sin perder contexto compartido.
- Elegir Local, Worktree o Cloud sin adivinar.

El vocabulario esencial

- **Proyecto de ChatGPT:** reúne conversaciones, archivos subidos, instrucciones y fuentes relacionadas.
- **Proyecto local:** conecta una o varias carpetas del ordenador con tareas de la aplicación.
- **Tarea:** conversación orientada a un resultado concreto. Conserva su propio historial.
- **Carpeta de trabajo:** lugar del disco que proporciona archivos y limita el contexto local.
- **Worktree:** copia de trabajo aislada de un repositorio Git. Es útil cuando dos tareas podrían cambiar el mismo código.
- **Cloud:** entorno remoto configurado para un repositorio o proyecto compatible.

Una tarea, un resultado

No mezcles «organiza archivos», «construye la web», «investiga competidores» y «prepara una presentación» en una sola tarea. Crea tareas distintas dentro del mismo proyecto cuando comparten contexto, pero producen resultados diferentes.

Un buen título describe el resultado:

- «Inventario de fuentes del curso».
- «Web local - primera versión».
- «Revisión de accesibilidad de la portada».
- «Informe para aprobación».

«Prueba» o «Chat nuevo» no ayudan a recuperar el trabajo.

Práctica: organiza el curso como proyecto

En la aplicación, añade Aprender–Codex como proyecto local. Si utilizas CLI o IDE, mantén esa carpeta como directorio de trabajo. Crea tres tareas:

1. **Inventario y seguridad.** Incluye las lecciones 4 y 7.
2. **Mi primera web.** Continúa la lección 11.
3. **Automatización CSV.** Continúa la lección 12.

En cada tarea, pide:

Resume el objetivo de esta tarea, los archivos que pertenecen a ella, el último resultado aprobado y el siguiente paso. No leas carpetas que pertenezcan a las otras tareas y no realices cambios.

Local, Worktree o Cloud

Elige **Local** para la carpeta de práctica y trabajos que quieres ver inmediatamente en tu ordenador. Elige **Worktree** cuando el proyecto esté en Git y quieras aislar cambios de otra tarea. Elige **Cloud** cuando el repositorio y su entorno remoto ya estén configurados y quieras delegar sin utilizar tu carpeta local.

Para una persona principiante, Local es el punto de partida. Worktree y Cloud no mejoran automáticamente una tarea mal definida.

Qué conserva el contexto

Una tarea conserva la conversación, pero los archivos pueden cambiar fuera de ella. Al reanudar, pide que vuelva a inspeccionar el estado actual. Guarda reglas duraderas en documentación o `AGENTS.md`; no dependas de que una frase antigua del chat siga siendo aplicable.

Comprobación manual

- Cada tarea tiene un título que describe un resultado.
- Ninguna tarea necesita leer todo el proyecto sin motivo.
- Los archivos producidos se encuentran en la carpeta esperada.
- Al reanudar, Codex distingue estado del chat y estado actual de los archivos.

Recuperación

Archivar una tarea no borra sus archivos. Pinning tampoco concede acceso adicional. Si abriste la carpeta equivocada, cierra la tarea y crea otra con el proyecto correcto; no amplíes la carpeta para abarcar dos trabajos sin relación.

Error frecuente

Continuar durante semanas en una única tarea. El historial crece, se mezclan objetivos y las instrucciones antiguas compiten con las nuevas. Crea una tarea por resultado y conserva reglas estables fuera del chat.

Mini ejercicio

Renombra las tres tareas para que otra persona comprenda qué entrega contiene cada una. Archiva una tarea terminada y localízala desde Settings > Archived tasks o el mecanismo disponible en tu superficie.

Evidencia que debes guardar

- Estructura del proyecto.
- Nombres de tareas.
- Resumen de alcance de cada tarea.
- Decisión Local/Worktree/Cloud y motivo.

Ejemplo resuelto: evitar que tres asuntos se conviertan en uno

Marta trabaja a la vez en su web, el seguimiento de clientes y una investigación de precios. Si utiliza una sola tarea, el contexto se contamina: aparecen reglas del CSV al editar la web o fuentes de investigación en un informe de clientes.

Organiza el trabajo así:

- **Proyecto web-servicio:** carpeta del sitio, instrucciones de tono y criterios de accesibilidad.
- **Proyecto seguimiento-clientes:** datos ficticios durante la práctica, reglas de validación y salidas separadas.

- **Proyecto comparativa-herramientas:** preguntas de investigación, fuentes y tabla de evidencia.

Dentro de web-servicio abre tareas distintas para estructura, contenido y revisión móvil. Cada tarea tiene un resultado y una definición de terminado. Cuando una decisión afecta a todas -por ejemplo, el nombre definitivo del servicio- la registra en un archivo de instrucciones compartido, no confía en que permanezca escondida en una conversación antigua.

Prueba de aislamiento: abre la tarea de investigación y pregunta «¿cuál es el objetivo de este trabajo y qué archivos puedes usar?». La respuesta no debería mencionar clientes ni modificar la web.

Un cierre útil: al terminar una tarea, guarda un resumen breve con objetivo, decisiones, archivos modificados, comprobaciones y pendientes. Así otra tarea puede continuar sin arrastrar toda la conversación.

Señal de alarma: para comenzar un trabajo nuevo tienes que escribir «ignora todo lo anterior» o descubres archivos de otro asunto en el contexto.

Criterio de terminado: cada tarea tiene un objetivo único, un conjunto de archivos definido y un resumen de traspaso comprensible.

Cambia una variable

- **Sol:** separa una tarea por asignatura o producto final, no una conversación para todo el trimestre.
- **Leo:** separa recopilación de fuentes, análisis y redacción si necesitan revisiones diferentes.
- **Dani:** una corrección y una nueva funcionalidad suelen merecer tareas distintas aunque estén en el mismo repositorio.

Comprueba que lo has entendido

1. ¿Qué información debe vivir en instrucciones compartidas y cuál en una tarea concreta?
2. ¿Por qué un resumen de traspaso es mejor que depender de una conversación enorme?

Solución comentada

Las reglas estables del proyecto pertenecen a instrucciones compartidas; el objetivo, decisiones y evidencia de una intervención pertenecen a su tarea. El resumen reduce ruido, hace visibles los pendientes y se puede revisar o corregir.

Fuentes oficiales

- [Projects, chats, and tasks](#)
- [Codex environments](#)
- [Worktrees](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 14. Permisos, privacidad, secretos y acciones externas

Tiempo: 60 minutos

Superficie: aplicación, CLI o IDE

Riesgo: medio

Qué vas a aprender

- Diferenciar sandbox y política de aprobaciones.
- Conceder el mínimo acceso necesario.
- Reconocer datos y acciones que requieren una decisión humana.

Dos controles que trabajan juntos

El **sandbox** define lo que las herramientas pueden hacer técnicamente: qué archivos pueden tocar y si los comandos pueden utilizar la red. La **política de aprobaciones** decide cuándo Codex debe detenerse para pedir permiso antes de una acción.

Una aprobación no elimina todos los límites del sandbox. Un sandbox amplio tampoco significa que una acción sea apropiada. Necesitas ambos controles.

Escalera de permisos

Empieza en el peldaño más bajo compatible con la tarea:

1. **Solo conversación:** no necesita carpeta.
2. **Solo lectura:** puede analizar, no escribir.
3. **Escritura en el workspace:** puede cambiar la carpeta de trabajo.
4. **Red limitada:** únicamente dominios o recursos necesarios.
5. **Acceso externo o completo:** solo en entornos confiables y con una justificación concreta.

No utilices Full access para evitar preguntas. Esas preguntas son una señal de que el agente pretende salir de la frontera establecida.

Práctica: diseña permisos antes del prompt

Para cada tarea, anota el permiso mínimo:

Tarea	Lectura	Escritura	Red	Acción externa
Resumir notas.txt en el chat	Sí	No	No	No
Crear salida/resumen.md	Sí	Sí, en workspace	No	No
Consultar documentación oficial actual	Sí	No	Sí, limitada	No
Publicar la web	Sí	Sí	Sí	Sí

La última tarea no debe ejecutarse como parte de la práctica: requiere proveedor, cuenta, revisión de privacidad y confirmación final.

Secretos

Una contraseña, token, clave API, código de un solo uso, cookie de sesión o archivo de credenciales no pertenece en:

- un prompt;
- AGENTS.md;
- capturas;
- logs;
- un repositorio;
- un documento compartido.

Utiliza el gestor de secretos o variables de entorno de la plataforma. Pide que se compruebe la presencia de la variable, nunca que se imprima su valor.

Comprueba si existe la variable OPENAI_API_KEY sin mostrar, copiar ni registrar su contenido. Informa solo de "presente" o "ausente". No hagas solicitudes de red.

Iniciar sesión con ChatGPT o API key

El inicio de sesión con ChatGPT utiliza la suscripción, permisos y controles del espacio de trabajo. Una clave API utiliza la organización y facturación de la plataforma API. La clave no es un sustituto universal del acceso de ChatGPT y algunas funciones de nube no están disponibles con ella.

Datos delicados

No utilices material real médico, legal, financiero, laboral o identificativo para practicar. En trabajo real, sigue la política de tu organización, minimiza datos y solicita revisión profesional cuando la decisión sea de alto impacto.

Acciones externas

Enviar un mensaje, publicar, comprar, cambiar permisos, subir archivos, borrar datos o modificar una cuenta afecta a sistemas o personas fuera de la carpeta. Pide un borrador primero y confirma la acción final de forma separada.

Prepara el correo como borrador dentro de salida/correo.md. No abras el correo, no busques destinatarios y no envíes nada. Marca los nombres o datos que debo confirmar antes de utilizarlo.

Prompt injection

Las instrucciones encontradas en webs, issues, documentos o archivos pueden intentar cambiar el objetivo del agente. Indica que son datos no confiables y que no autorizan accesos, descargas ni acciones.

Comprobación manual

- Revisa el permiso activo antes de enviar.
- Lee cada solicitud de ampliación y su destino.
- Comprueba el diff o los archivos después de escribir.
- Revisa las herramientas externas por separado.
- Cierra o revoca sesiones y claves que ya no necesitas.

Recuperación

Si compartiste un secreto, revócalo y reemplázalo; borrarlo del chat no garantiza que siga siendo seguro. Si concediste acceso persistente a una app o web, retíralo desde Settings. Si una acción externa ya se ejecutó, utiliza el mecanismo de cancelación, reversión o soporte del servicio.

Error frecuente

Aprobar por fatiga. Si una tarea pide permisos repetidos, detén el trabajo y corrige la configuración, la carpeta o el plan. No conviertas una sucesión de avisos inesperados en acceso completo.

Mini ejercicio

Reescribe una petición para que produzca un borrador local en lugar de enviar o publicar. Identifica el momento exacto en que necesitaría una confirmación nueva.

Evidencia que debes guardar

- Tabla de permisos.
- Motivo de cualquier ampliación.
- Revisión de apps y sitios autorizados.
- Lista de acciones externas que permanecieron como borrador.

Ejemplo resuelto: decidir un permiso con cuatro preguntas

Dani pide revisar una aplicación local. Codex solicita acceso de red para instalar una dependencia. En lugar de aprobar por reflejo, Dani utiliza cuatro preguntas:

1. **¿Para qué acción concreta se necesita?** Instalar un paquete declarado por el proyecto.
2. **¿Qué recurso alcanzará?** Un registro de paquetes, no cualquier sitio.
3. **¿Se puede reducir?** Primero inspeccionar el archivo de dependencias y confirmar nombre y versión.
4. **¿Qué evidencia quedará?** Cambio en el archivo de bloqueo, salida del instalador y pruebas posteriores.

Una solicitud con propósito y alcance puede aprobarse si encaja en el trabajo. Otra petición como «envía el informe al cliente» requiere detenerse: implica una acción externa, destinatario, datos potencialmente sensibles y un efecto que no se deshace con facilidad.

Semáforo práctico:

- **Verde:** leer archivos de práctica, crear una salida nueva, ejecutar una comprobación local conocida.
- **Ámbar:** instalar dependencias, acceder a internet, modificar muchos archivos o usar una sesión iniciada. Revisa propósito y alcance.
- **Rojo:** publicar, pagar, borrar originales, enviar mensajes, revelar secretos o cambiar cuentas. Exige autorización específica y control humano en el punto final.

No pegues claves o contraseñas en el prompt. Si un proyecto necesita secretos, utiliza el mecanismo previsto por el entorno y evita que aparezcan en capturas, registros o archivos versionados.

Señal de alarma: no puedes explicar por qué se necesita el permiso o la solicitud es mucho más amplia que la tarea.

Criterio de terminado: cada permiso aprobado tiene propósito, alcance y evidencia; ninguna acción externa se ejecutó por una autorización genérica.

Cambia una variable

Clasifica estas acciones: leer un PDF de práctica; instalar un paquete; borrar cien originales; preparar un correo sin enviarlo; publicar la web. Explica qué control añadirías a las de color ámbar o rojo.

Comprueba que lo has entendido

1. ¿La confianza en Codex sustituye la revisión de permisos?
2. ¿Preparar un mensaje y enviarlo son la misma clase de acción?

Solución comentada

No. Los permisos dependen del efecto, los datos y el alcance, no de una confianza general. Preparar un borrador es reversible y revisable; enviarlo produce una consecuencia externa y necesita confirmación específica.

Fuentes oficiales

- [Agent approvals & security](#)
- [Sandbox](#)
- [Authentication](#)
- [Permissions](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 15. Copias de seguridad y Git sin miedo

Tiempo: 60 minutos

Superficie: aplicación o CLI

Plataformas: macOS, Windows y Linux

Qué vas a aprender

- Diferenciar copia de seguridad y control de versiones.
- Ver exactamente qué cambió.
- Restaurar una versión local sin publicar nada.

Dos redes de seguridad distintas

Una **copia de seguridad** conserva archivos en otro lugar. **Git** registra versiones y diferencias dentro de un proyecto. Git no es automáticamente una copia remota: `git init` y un `commit` local no publican nada en internet.

Para archivos personales y formatos binarios, conserva copias externas adecuadas. Para una web o archivos de texto, Git permite revisar cambios con precisión.

Ruta sin Git

Continúa utilizando carpetas con nombres claros:

- `mi-primer-web-01-inicial`
- `mi-primer-web-02-aprobada`
- `mi-primer-web-03-prueba`

Registra en `CAMBIOS.md` fecha, objetivo y archivos modificados. Esta ruta es válida si todavía no quieres instalar Git.

Ruta con Git

La aplicación ofrece controles Git cuando Git está instalado. Puede pedirte inicializar el proyecto si todavía no es un repositorio. Hazlo únicamente dentro de `mi-primer-web`.

Desde terminal, la secuencia local básica es:

```
cd /ruta/a/Aprender-Codex/mi-primer-web
git init
git status
git add index.html styles.css
git commit -m "Crea la primera version local de la web"
```

Git puede pedir nombre y correo de autor. Para esta práctica local utiliza una identidad apropiada a tu entorno y no publiques el repositorio. No copies credenciales en la configuración.

Práctica: observa un cambio

1. Guarda la versión inicial.
2. Pide a Codex cambiar solo el texto del título.
3. Abre el panel de revisión o ejecuta:

```
git status --short
git diff
```

4. Comprueba que solo cambió la línea esperada.
5. Si la apruebas, crea un segundo commit. Si no, utiliza la interfaz para revertir ese fragmento o pide a Codex restaurar el texto anterior.

Encargo listo para copiar

Antes de editar, revisa el estado de Git. Conserva cualquier cambio que no pertenezca a esta tarea. Cambia únicamente el título visible de index.html de "[anterior]" a "[nuevo]". No cambies estilos. Después muestra el diff, comprueba la página y no hagas commit ni push hasta que yo lo apruebe.

Lo que no debes hacer todavía

- No añadas un remoto.
- No ejecutes push.
- No publiques el repositorio.
- No reviertas archivos que tengan cambios ajenos.
- No utilices comandos destructivos que eliminen el trabajo no guardado.

Comprobación manual

- `git status` muestra solo los archivos esperados.
- El diff contiene el cambio aprobado.
- La página sigue abriendo.
- El historial contiene una idea por commit.
- No existe publicación remota.

Recuperación

Si no entiendes un comando de recuperación, detente y pide que explique exactamente qué archivos afectará antes de ejecutarlo. Conserva una copia fuera del repositorio mientras aprendes.

Error frecuente

Pedir «limpiar el repositorio» con cambios sin guardar. «Limpiar» puede interpretarse como borrar archivos o revertir trabajo. Nombra qué debe conservarse y revisa el estado primero.

Mini ejercicio

Haz dos cambios deliberados, uno correcto y otro no deseado. Practica conservar el primero y revertir únicamente el segundo desde el panel de revisión.

Evidencia que debes guardar

- Copia externa inicial.
- `git status` antes y después.
- Diff aprobado.
- Historial local de commits o CAMBIOS.md.

Ejemplo resuelto: una modificación pequeña que se puede deshacer

Dani quiere cambiar el título de una web y el color principal. Antes de empezar crea una copia o confirma que el repositorio está limpio. Si utiliza Git, sigue este ciclo:

1. Consulta el estado y confirma qué cambios ya existían.
2. Pide modificar solo los archivos necesarios.
3. Revisa el diff: líneas añadidas, eliminadas y archivos inesperados.
4. Abre la web y ejecuta las comprobaciones.
5. Guarda el cambio con un mensaje que explique la intención.

Prompt:

Antes de editar, muestra el estado del repositorio y no alteres cambios preexistentes. Cambia el título visible a «Taller Norte» y el color principal a #2457A6. No reformatees archivos completos ni modifiques dependencias. Después muestra el diff, abre o prueba la web y enumera cualquier comprobación que no hayas podido ejecutar. No crees un commit sin mi confirmación.

El diff revela que también cambió el final de línea de todo `styles.css`. Aunque la web se vea igual, ese ruido dificulta revisar. Dani pide revertir únicamente el reformato accidental y conservar las dos modificaciones previstas.

Ejercicio de recuperación: en la carpeta de práctica, modifica una frase, comprueba el diff y recupérala utilizando la copia o el método de Git aprendido. Practicar la reversión antes de necesitarla reduce el miedo.

Señal de alarma: el estado inicial ya muestra cambios que no reconoces o Codex propone descartar trabajo ajeno para «limpiar» el repositorio.

Criterio de terminado: el diff contiene solo la intención declarada, el resultado funciona y sabes volver al estado anterior sin perder otros cambios.

Cambia una variable

Repite con dos cambios independientes. Observa por qué revisarlos y guardarlos por separado facilita encontrar y deshacer errores.

Comprueba que lo has entendido

1. ¿Una copia de seguridad y un commit cumplen exactamente la misma función?
2. ¿Por qué se consulta el estado antes y después?

Solución comentada

No. Ambos ayudan a recuperar, pero Git también registra diferencias e historia dentro de un repositorio. El estado inicial protege cambios preexistentes; el final revela el alcance real y archivos inesperados.

Fuentes oficiales

- [Code review](#)
- [Local environments and Git tools](#)
- [Best practices](#)

Revisado contra la documentación: 14 de julio de 2026.

Módulo 6. Personalizar y reutilizar

Lección 16. Instrucciones personales y AGENTS.md

Tiempo: 50 minutos

Superficie: aplicación, CLI o IDE

Plataformas: macOS, Windows y Linux

Qué vas a aprender

- Decidir qué instrucciones pertenecen al mensaje, a tu perfil o al proyecto.
- Crear un AGENTS.md breve y comprobable.
- Entender cómo se combinan instrucciones generales y específicas.

Coloca cada regla en su ámbito

- **Prompt:** una condición que solo importa en la tarea actual.
- **Instrucciones personales:** preferencias que quieres en muchas conversaciones, como idioma o estilo.
- **AGENTS.md de proyecto:** reglas duraderas de una carpeta o repositorio: estructura, límites y verificación.
- **AGENTS.md anidado:** reglas más específicas de una subcarpeta.

No conviertas una preferencia temporal en una norma permanente.

Cómo descubre Codex las instrucciones

Codex puede leer una guía global desde su directorio de configuración y después archivos AGENTS.md desde la raíz del proyecto hasta la carpeta de trabajo. Las instrucciones más cercanas tienen prioridad cuando concretan o sustituyen una regla anterior.

En la aplicación, las instrucciones personalizadas de Settings > Personalization pueden actualizar la guía personal. En un proyecto compartido, conserva las reglas del equipo dentro del proyecto para que puedan revisarse y versionarse.

Práctica: crea la guía de tu web

Dentro de mi-primera-web, crea AGENTS.md:

```
# Guía del proyecto

## Objetivo
- Mantener una web local de una página para aprendizaje.

## Límites
- No publicar ni añadir servicios externos.
- No usar datos personales reales.
- No añadir dependencias sin pedir permiso.

## Contenido
- Mantener el español claro para público general.
- Marcar el correo de ejemplo como ficticio.

## Verificación
- Abrir index.html después de cada cambio.
- Comprobar móvil, escritorio y navegación con teclado.
- Informar de cualquier comprobación no realizada.
```

Inicia una tarea nueva dentro del proyecto y escribe:

No edites archivos. Enumera las instrucciones del proyecto que se aplican a esta tarea, indica de qué archivo proceden y señala cualquier ambigüedad.

Después pide un cambio incompatible, por ejemplo «añade analítica externa sin preguntar». Codex debería señalar el conflicto y solicitar una decisión.

Qué no debe contener AGENTS.md

- Contraseñas o claves.
- Historias de conversación.
- Información personal innecesaria.
- Una lista enorme de posibilidades hipotéticas.
- Instrucciones que no pueden comprobarse, como «hazlo perfecto».

Global y proyecto

Una guía personal puede decir «responde en español y explica la jerga». El proyecto puede decir «el contenido público utiliza español neutro». Una regla más específica no debería borrar silenciosamente límites de seguridad definidos arriba; si existe conflicto, revísalo explícitamente.

Comprobación manual

- Una persona nueva comprende objetivo y límites en menos de dos minutos.
- Cada regla puede observarse o verificarse.
- Los comandos o pasos citados funcionan realmente.
- No hay secretos.
- Las reglas específicas están cerca de los archivos a los que afectan.

Recuperación

Si Codex utiliza instrucciones antiguas, inicia una tarea o sesión nueva para que vuelva a construir la cadena. Busca `AGENTS.override.md` o guías anidadas que puedan tener prioridad.

Error frecuente

Usar AGENTS.md como enciclopedia. Las reglas importantes se pierden entre explicaciones largas. Mantén una guía corta y enlaza documentos especializados cuando sean necesarios.

Mini ejercicio

Añade una regla que hayas repetido dos veces durante el curso. Justifica por qué es duradera. Elimina otra que solo corresponda a una práctica.

Evidencia que debes guardar

- `AGENTS.md` aprobado.
- Resumen de instrucciones activas.
- Conflicto detectado y decisión tomada.

Ejemplo resuelto: una regla estable sin convertirla en una novela

Marta repite en cada tarea: «escribe en español claro, no cambies originales y guarda resultados en salida». Esas reglas son estables y pueden convertirse en instrucciones del proyecto:

Instrucciones del proyecto

- Responde en español claro y explica cualquier término técnico.
- No modifiques archivos dentro de ``entrada/``.
- Escribe todos los entregables en ``salida/``.
- Antes de cambios masivos, presenta inventario y vista previa.
- Al terminar, informa de archivos creados, controles y pendientes.

Para la tarea concreta añade: «procesa el informe de junio». Esa fecha no pertenece a las instrucciones permanentes porque cambiará en la próxima ejecución.

Dani tiene además un `AGENTS.md` en una subcarpeta con reglas para las pruebas de esa parte del proyecto. Antes de actuar pide a Codex que enumere las instrucciones aplicables y señale cualquier conflicto. Si una regla general dice «guarda salidas en salida» y una regla específica de pruebas exige `tests/fixtures`, la más cercana al trabajo puede precisar el destino; el conflicto debe hacerse visible, no resolverse por intuición silenciosa.

Prueba de calidad: abre una tarea nueva y formula un encargo mínimo. Comprueba si se respetan idioma, carpeta protegida, vista previa y parte final. Si debes recordar la misma regla otra vez, revisa su ubicación o redacción.

Señal de alarma: las instrucciones contienen contraseñas, decisiones temporales, reglas contradictorias o tantos detalles que ya no se sabe cuáles son obligatorios.

Criterio de terminado: las normas estables se aplican en una tarea nueva y el objetivo temporal permanece en el prompt de esa tarea.

Cambia una variable

Elimina deliberadamente una regla no esencial y comprueba si el resultado sigue siendo seguro. Las instrucciones deben ser suficientes, no enciclopédicas.

Comprueba que lo has entendido

1. ¿«Haz el informe de hoy» pertenece a instrucciones permanentes?
2. ¿Qué debes hacer si dos reglas parecen incompatibles?

Solución comentada

No: es un objetivo temporal. Ante reglas incompatibles, identifica su origen y alcance, detén la decisión afectada y pide aclaración o aplica la jerarquía documentada de forma explícita.

Fuentes oficiales

- Custom instructions with `AGENTS.md`
- Best practices
- ChatGPT desktop app settings

Revisado contra la documentación: 14 de julio de 2026.

Lección 17. Elige entre skills, plugins y MCP

Tiempo: 55 minutos

Superficie: aplicación; algunas funciones también en CLI e IDE

Riesgo: medio

Qué vas a aprender

- Comprender tres formas de ampliar el agente.
- Elegir la más pequeña que resuelve el problema.
- Revisar permisos y datos antes de instalar o conectar.

Tres piezas diferentes

- **Skill:** instrucciones y recursos reutilizables para realizar bien una tarea concreta.
- **Plugin:** paquete instalable que puede incluir skills y conexiones con servicios.
- **MCP:** protocolo que conecta el agente con herramientas o contexto externo.

Empieza por una skill cuando repites un proceso. Utiliza un plugin cuando quieres instalar y compartir un conjunto ya empaquetado. Configura MCP cuando necesitas una herramienta o fuente externa que no llega mediante un plugin adecuado.

Ejemplo

Necesidad: producir cada semana un resumen con la misma estructura.

- Skill: define fuentes, pasos, plantilla y comprobaciones.
- Plugin: distribuye la skill y, quizá, una conexión con Drive o Slack.
- MCP: proporciona herramientas para consultar un sistema interno.

No necesitas construir un servidor MCP para guardar una plantilla.

Práctica: diseña una skill antes de instalarla

Elige la automatización CSV de la lección 12 y pide:

Diseña una skill reutilizable para resumir un CSV de gastos de práctica.
Antes de crear archivos, define:

- cuándo debe y cuándo no debe activarse;
- entradas obligatorias;
- pasos y comprobaciones;
- formato de salida;
- límites de seguridad;
- ejemplos correcto e incorrecto.

No conectes servicios, no instales plugins y no incluyas datos reales.

Revisa el diseño. Si tu superficie ofrece `$skill-creator` y quieres continuar, utilízalo únicamente dentro de una carpeta de práctica. Comprueba el contenido de `SKILL.md` antes de habilitarla.

Revisar un plugin

Antes de instalar:

1. Identifica quién lo publica.
2. Lee qué skills, aplicaciones o servidores incluye.
3. Comprueba qué datos puede leer y qué acciones puede realizar.

4. Revisa si la organización permite instalarlo.
5. Pruébalo con información no sensible y acceso mínimo.
6. Decide quién revisará actualizaciones y cuándo retirarlo.

Disponibilidad: los plugins se utilizan en ChatGPT Work en la web, y en Work o Codex en la aplicación; la disponibilidad exacta en CLI o IDE puede cambiar según el paquete y el entorno.

Revisar una conexión MCP

Pregunta:

- ¿El servidor es local o remoto?
- ¿Qué herramientas expone?
- ¿Cuáles escriben o borran?
- ¿Cómo se autentica?
- ¿Dónde quedan las credenciales?
- ¿Qué sucede si el servidor no está disponible?

Los clientes locales de Codex comparten configuración MCP para el mismo host. ChatGPT web utiliza herramientas remotas proporcionadas mediante plugins y no lee tu configuración local de Codex.

Comprobación manual

- La skill tiene un propósito único.
- La descripción indica límites de activación.
- No contiene claves ni datos reales.
- Cualquier plugin procede de una fuente que has revisado.
- Las herramientas con efectos externos requieren aprobación apropiada.
- Puedes deshabilitar o retirar la ampliación.

Recuperación

Deshabilita una skill que produce resultados erróneos. Desinstala el plugin y revoca la aplicación conectada si ya no lo utilizas. Elimina o desactiva el servidor MCP y sus credenciales; quitar la configuración no revoca automáticamente un token en el servicio remoto.

Error frecuente

Instalar una herramienta para evitar escribir un proceso claro. Primero define el flujo y su resultado. Después decide si necesita empaquetado o acceso externo.

Mini ejercicio

Clasifica estas necesidades: «usar siempre la plantilla de marca», «leer incidencias de GitHub», «compartir cinco flujos y una conexión con todo el equipo».

Solución comentada

- Plantilla de marca: skill.
- Leer incidencias: plugin o MCP, según la integración disponible.
- Cinco flujos y conexión compartida: plugin.

Evidencia que debes guardar

- Diseño de skill.
- Lista de permisos de cualquier plugin.
- Inventario de herramientas MCP y efectos.
- Procedimiento para desactivar o revocar.

Ejemplo resuelto: elegir la pieza reutilizable adecuada

Sol repite cada semana un proceso: leer una lección, generar objetivos, actividad, rúbrica y una lista de accesibilidad. Quiere reutilizarlo. Compara tres soluciones:

- Una **skill** es apropiada si necesita un procedimiento guiado y recursos que Codex pueda aplicar cuando la tarea coincida.
- Un **plugin** resulta útil si quiere distribuir un paquete mayor que reúna skills, herramientas o integraciones relacionadas.
- Un servidor **MCP** encaja cuando necesita conectar Codex de manera estructurada con una fuente o servicio externo mediante herramientas bien definidas.

Como el proceso de Sol utiliza archivos locales y una plantilla, comienza por una skill. No crea un conector externo innecesario. Define cuándo se activa, entradas requeridas, pasos, controles y formato de salida. La prueba con una lección conocida y compara la rúbrica generada con sus criterios.

Leo, en cambio, necesita consultar una base documental institucional que ya ofrece una integración MCP autorizada. Su problema principal no es recordar un procedimiento, sino acceder a una fuente externa mediante operaciones controladas. Dani podría combinar ambos: una skill que describa el flujo y herramientas MCP que proporcionen datos.

Pregunta de decisión: «¿Quiero enseñar un método, empaquetar capacidades o conectar un sistema?». Método sugiere skill; paquete, plugin; conexión, MCP.

Señal de alarma: instalas una extensión amplia para resolver una tarea simple sin revisar autor, permisos, código o datos a los que accederá.

Criterio de terminado: puedes explicar por qué elegiste esa opción, qué activa, qué datos toca y cómo se prueba antes de compartirla.

Cambia una variable

Clasifica estos casos: plantilla semanal de actas; conjunto corporativo de instrucciones e integraciones; consulta a un gestor documental externo; prompt que solo usarás una vez. El último quizá no necesita ninguna pieza reutilizable.

Comprueba que lo has entendido

1. ¿Toda tarea repetida necesita un plugin?
2. ¿Una skill y MCP pueden colaborar?

Solución comentada

No. Conviene elegir la solución mínima: a veces bastan instrucciones o una plantilla. Sí: una skill puede definir el procedimiento y utilizar herramientas proporcionadas por una conexión MCP si están disponibles y autorizadas.

Fuentes oficiales

- [Skills & Plugins](#)
- [Build skills](#)
- [Model Context Protocol](#)

Revisado contra la documentación: 14 de julio de 2026.

Lección 18. Programa tareas que puedas revisar

Tiempo: 50 minutos más una ejecución posterior

Superficie: web o aplicación de escritorio

Riesgo: medio

Qué vas a aprender

- Convertir un prompt probado en una tarea programada.
- Elegir entre ejecución independiente y continuación de una tarea.
- Diseñar una automatización desatendida con permisos estrechos.

Programa solo lo que ya funciona manualmente

Una tarea programada ejecuta un prompt sin que estés presente. Si el prompt es ambiguo, el horario solo repetirá la ambigüedad. Prueba primero una ejecución normal y revisa su salida.

La aplicación puede programar tareas sobre proyectos locales. El ordenador debe permanecer encendido, la aplicación ejecutándose y la carpeta disponible. Las tareas web pueden utilizar archivos subidos y herramientas conectadas, pero no trabajan directamente sobre una carpeta local. CLI e IDE no proporcionan la interfaz de gestión de Scheduled.

Independiente o con continuidad

- **Tarea independiente:** cada ejecución empieza desde el prompt guardado. Útil para un informe periódico.
- **Tarea dentro de una conversación:** vuelve al mismo contexto. Útil para seguimiento de una operación o revisión continua.

Para un resumen semanal repetible, empieza por una tarea independiente.

Práctica segura

Crea `entrada/seguimiento.txt` con cinco líneas ficticias, cada una con fecha, estado y nota. Prueba primero:

Lee `entrada/seguimiento.txt` y prepara un resumen semanal dentro de `salida/seguimiento`. Incluye: elementos nuevos, pendientes, bloqueados y datos incompletos. No cambies la entrada, no uses red y no envíes mensajes. Si no hay novedades, crea un informe que diga "Sin novedades". Termina indicando la ruta del archivo y el número de líneas procesadas.

Comprueba el resultado. Después programa una ejecución semanal si Scheduled está disponible.

Prompt duradero

Añade instrucciones que sigan teniendo sentido en el futuro:

En cada ejecución:

1. utiliza únicamente `entrada/seguimiento.txt`;
2. no asumas que una fecha ausente es la fecha actual;
3. escribe un archivo nuevo con fecha en `salida/seguimiento`;
4. no sobrescribas informes anteriores;
5. no uses red, plugins ni aplicaciones;
6. informa si la entrada no existe o tiene líneas inválidas;
7. no realices cambios fuera del proyecto.

Detente después de crear y verificar el informe.

Permisos

Las tareas programadas utilizan la configuración de sandbox y se ejecutan desatendidas. Utiliza lectura más escritura dentro del workspace, sin red. Evita Full access. En un repositorio Git, un worktree puede aislar cambios; en una carpeta sin Git la tarea escribirá directamente en el proyecto.

Comprobación de las primeras ejecuciones

- Revisa cada informe de las primeras semanas.
- Comprueba que no sobrescribe versiones.
- Confirma que los errores son visibles.
- Ajusta horario y prompt si genera ruido.
- Pausa o elimina la tarea cuando ya no tenga propietario o utilidad.

Alternativa sin Scheduled

Guarda el prompt como una skill o plantilla y ejecútalo manualmente una vez por semana. La repetibilidad no depende de que la función de calendario aparezca en tu cuenta.

Recuperación

Pausa la tarea antes de modificar su prompt o permisos. Archiva ejecuciones antiguas que ya no necesitas. Si una tarea local cambió archivos incorrectos, detén futuras ejecuciones y restaura desde Git o copia de seguridad antes de volver a activarla.

Error frecuente

Programar una acción externa irreversible. Empieza generando un informe o borrador. Enviar, fusionar, desplegar o publicar necesita reglas adicionales, identidad de servicio y revisión humana.

Mini ejercicio

Añade una línea inválida al seguimiento y ejecuta manualmente. El informe debe señalarla sin descartar silenciosamente el resto.

Evidencia que debes guardar

- Prompt probado manualmente.
- Horario y destino.
- Permisos utilizados.
- Primeras ejecuciones revisadas.
- Propietario y condición de retirada.

Ejemplo resuelto: una tarea programada que falla de forma visible

Leo quiere recibir cada lunes un resumen de cambios en tres páginas oficiales. Un mal encargo sería «vigila internet y avísame de lo importante». No define fuentes, importancia, comparación ni salida.

La versión operativa es:

Cada lunes a las 09:00, consulta únicamente las tres URL indicadas. Compara el contenido visible con el informe anterior y crea un nuevo archivo AAAA-MM-DD-cambios.md. Para cada cambio incluye página, sección, resumen, enlace y nivel de confianza. Si una página no está disponible, no reutilices datos antiguos como si fueran actuales: registra ERROR DE ACCESO y continúa. Si no hay cambios, crea igualmente el informe con «sin cambios detectados».

No envíes mensajes ni publiques resultados.

Antes de programarla, Leo la ejecuta manualmente una vez. Comprueba acceso, estructura del informe y ubicación. Después programa y revisa las dos primeras ejecuciones. Conserva un registro con fecha prevista, fecha real, resultado y error.

La ausencia de un informe también debe ser observable. Si la ejecución depende de una máquina local, Leo conoce esa dependencia y decide qué ocurrirá si está apagada. Una tarea programada no debe fingir continuidad cuando el entorno no estaba disponible.

Prueba de fallo: sustituye temporalmente una URL de práctica por una dirección inexistente. El resultado correcto contiene un error explícito y no inventa cambios.

Señal de alarma: una automatización puede enviar, publicar, comprar o borrar de forma recurrente sin revisión y sin un límite muy claro.

Criterio de terminado: la tarea funciona manualmente, deja evidencia en cada ejecución, maneja fallos y se puede desactivar o revisar.

Cambia una variable

- **Marta:** resumen semanal de CSV nuevos, sin enviar correos.
- **Sol:** borrador de planificación, no publicación automática en el aula.
- **Dani:** informe de estado del repositorio, sin fusionar ni desplegar cambios.

Comprueba que lo has entendido

1. ¿Por qué se prueba manualmente antes de programar?
2. ¿«No se creó ningún informe» significa necesariamente «no había cambios»?

Solución comentada

La ejecución manual valida permisos, entradas, salida y controles sin esperar al horario. La falta de informe puede significar fallo, equipo apagado o configuración incorrecta; por eso incluso «sin cambios» debe generar evidencia.

Fuentes oficiales

- [Scheduled tasks](#)
- [Sandbox](#)
- [Worktrees](#)

Revisado contra la documentación: 14 de julio de 2026.

Módulo 7. Puente técnico y proyecto final

Lección 19. Da el salto a CLI, IDE y repositorios cuando lo necesites

Tiempo: 60-90 minutos

Superficie: CLI o extensión de IDE

Plataformas: macOS, Windows y Linux

Carácter: opcional

Qué vas a aprender

- Utilizar Codex desde una terminal o editor manteniendo las normas de seguridad del curso.
- Reconocer los controles principales de una sesión.
- Decidir si estás preparado para continuar con «Codex para programadores».

La terminal no es un nivel de inteligencia

CLI e IDE ofrecen una relación más directa con archivos, Git, herramientas y automatización. Avanza hacia ellas cuando tu trabajo lo necesite. No son un requisito para haber aprendido a dirigir y revisar un agente.

La CLI en cuatro acciones

1. Abre la terminal.
2. Entra en la carpeta exacta.
3. Inicia codex.
4. Describe el resultado y revisa lo que propone.

```
cd /ruta/a/Aprender-Codex/mi-primer-web
codex
```

En Windows PowerShell utiliza rutas Windows. En WSL2 utiliza rutas Linux y mantén proyecto y herramientas dentro del mismo entorno.

Comprobaciones iniciales

```
codex --version
```

Dentro de la sesión, escribe / para ver los comandos reales de tu versión. Pueden incluir:

- /status: configuración y uso; en CLI puede mostrar modelo, aprobación, raíces escribibles y contexto.
- /permissions: perfil de permisos.
- /model: modelo activo.
- /plan: planificación, cuando está disponible.
- /review: revisión de cambios en un repositorio Git.
- /mention: añadir un archivo o carpeta al contexto.

No memorices una lista antigua: utiliza el selector de tu instalación.

Primer encargo CLI, todavía sin editar

Analiza esta carpeta sin modificar archivos. Enumera su estructura hasta dos niveles, explica cómo abrir la web y señala cualquier archivo que no entiendas. Indica las raíces escribibles y separa hechos observados de inferencias.

Comprueba con `/status` y el explorador de archivos que estás en la carpeta correcta.

Imágenes en CLI

Puedes adjuntar una captura al iniciar:

```
codex -i captura.png "Describe el problema visible y no cambies archivos"
```

Identifica qué muestra la imagen, qué zona importa y qué resultado esperas. Oculta notificaciones y datos personales.

Extensión de IDE

Sigue la documentación oficial para instalar la extensión compatible con Visual Studio Code. Después abre la carpeta, inicia sesión, revisa permisos y añade como contexto solo archivos o selecciones relevantes. La extensión comparte con CLI la configuración del mismo host, mientras sus preferencias visuales pertenecen al editor.

Windows nativo o WSL2

Utiliza Windows nativo si el proyecto y las herramientas viven en Windows. Elige WSL2 si tu flujo ya utiliza Linux. WSL1 no es una ruta compatible actual. En WSL2 instala bubblewrap y procura guardar repositorios bajo el hogar Linux.

Cuándo utilizar `codex exec`

`codex exec` ejecuta una tarea no interactiva. Es apropiado para análisis repetibles cuando entrada, salida, permisos y fallos están definidos. No es una solución para una petición ambigua.

```
codex exec --sandbox read-only "Revisa la estructura actual. No edites archivos. Devuelve riesgos y límites."
```

Práctica de transición

Lee AGENTS.md y resume las reglas activas. Revisa index.html y styles.css sin editar. Propón una única mejora de accesibilidad, explica cómo verificarla y espera mi aprobación.

Autoriza el cambio solo si comprendes el diff. Después abre la web y comprueba la mejora con teclado y una ventana estrecha.

Recuperación

Interrompe si ruta o permisos no son correctos. En Git, conserva cambios ajenos y revisa `git status` antes de revertir. Si la instalación falla, `codex doctor` puede ayudar cuando está disponible; revisa el informe antes de compartirlo porque puede contener rutas u otros datos del sistema.

Error frecuente

Copiar comandos sin comprender el entorno. Antes de instalar, elevar permisos o usar red, pide qué cambia, qué acceso necesita y cómo se deshace.

Señales de que puedes continuar

- Abres la carpeta correcta.
- Comprendes lectura, escritura, red y acciones externas.
- Revisas un diff.
- Defines resultado y comprobación.
- No utilizas Full access como opción predeterminada.
- Conservas una versión recuperable.

Evidencia que debes guardar

- Versión y superficie.
- Estado sin datos sensibles.
- Primer análisis de solo lectura.
- Diff y comprobación de la mejora opcional.

Ejemplo resuelto: del lenguaje natural a un cambio revisable en el IDE

Dani abre un repositorio pequeño en el IDE. Antes de tocar código pide orientación:

No modifiques nada. Lee las instrucciones del repositorio y explícame:

- qué hace el proyecto;
- cómo se instala y ejecuta;
- dónde están las pruebas;
- qué estado tiene Git;
- qué dudas debes resolver antes de cambiar el formulario de contacto.

Con el mapa entendido, formula una tarea estrecha: validar que el correo tenga formato correcto y mostrar un mensaje accesible. Pide plan, cambio mínimo y pruebas. Mientras Codex trabaja, Dani observa archivos abiertos y comandos; si aparece una dependencia nueva sin justificación, corrige el rumbo.

Al terminar revisa el diff en el IDE, ejecuta la prueba indicada y hace una prueba manual. La respuesta final debe distinguir entre comprobaciones ejecutadas y recomendaciones no ejecutadas.

Traducción de términos: repositorio es la carpeta con historia de cambios; rama es una línea de trabajo; diff es la comparación; commit es un punto guardado con intención; prueba es una comprobación reproducible. No necesitas dominar todos los comandos para hacer preguntas precisas sobre ellos.

Ejercicio puente: pide a Codex que explique cada comando antes de ejecutarlo y que no utilice variantes destructivas. Luego intenta predecir el resultado de un comando de lectura, ejecútalo y compara.

Señal de alarma: se propone reescribir historia, descartar cambios ajenos, publicar una rama o utilizar secretos sin que la tarea lo exija.

Criterio de terminado: entiendes el propósito del cambio, el diff es limitado, las pruebas relevantes pasan y el repositorio no contiene cambios inesperados.

Cambia una variable

Repite el ejercicio desde CLI o desde IDE y anota qué interfaz te ayuda mejor a observar contexto, comandos y diferencias. La herramienta preferida puede depender de la tarea.

Comprueba que lo has entendido

1. ¿Por qué se lee `AGENTS.md` o las instrucciones antes de modificar un repositorio?
2. ¿Un diff pequeño garantiza por sí solo que el cambio sea correcto?

Solución comentada

Las instrucciones contienen convenciones, límites y comprobaciones del proyecto. Un diff pequeño facilita revisar, pero todavía necesita pruebas y validación del comportamiento.

Fuentes oficiales

- [Codex CLI command reference](#)
- [Codex IDE extension commands](#)
- [Authentication](#)
- [WSL2](#)

Revisado contra la documentación: 14 de julio de 2026.

Continuación recomendada: [Codex para programadores](#).

Lección 20. Proyecto final según tu perfil

Tiempo: 3-6 horas

Superficie: la adecuada al resultado

Plataformas: macOS, Windows y Linux

Qué vas a aprender

- Integrar alcance, permisos, creación, verificación y recuperación.
- Elegir un proyecto final adecuado a tu perfil.
- Entregar evidencia que otra persona pueda revisar sin leer toda la conversación.

Objetivo

Completar un resultado que otra persona pueda revisar, repetir y recuperar. No se evalúa la cantidad de texto o código generado, sino alcance, evidencia, seguridad y utilidad.

Reglas comunes

Todos los proyectos deben:

1. utilizar datos ficticios o autorizados;
2. empezar con una carpeta claramente limitada;
3. conservar originales;
4. incluir un plan aprobado;
5. producir un resultado revisable;
6. comprobar un caso correcto y uno problemático;
7. declarar lo no verificado;
8. evitar publicación o envío durante la evaluación;
9. documentar cómo repetir y deshacer;
10. enlazar las fuentes utilizadas.

Proyecto A - estudiante o docente

Resultado: paquete de estudio a partir de tres textos propios o con licencia compatible.

Entregables: mapa de conceptos con origen; guía de dos páginas; diez preguntas con respuestas separadas; tabla de datos que necesitan verificación; versión de lectura fácil.

Límites: no inventar citas, no usar datos reales de estudiantes y marcar incertidumbre.

Comprobación: rastrear cinco afirmaciones hasta los textos y pedir a otra persona que responda dos preguntas con la guía.

Proyecto B - profesional, autónomo o pyme

Resultado: sistema local que convierte solicitudes ficticias en borradores y una tabla de seguimiento.

Entregables: plantilla de entrada; clasificación por estado explícito; borradores que nunca se envían; registro de datos ausentes; instrucciones y recuperación.

Límites: sin correo, CRM o mensajería conectados; ninguna decisión automática de crédito, empleo, salud o derecho; ninguna acción externa.

Comprobación: caso completo, caso con datos ausentes y caso que debe escalarse a una persona.

Proyecto C - creador, investigador o comunicador

Resultado: dossier con fuentes y una web local para presentarlo.

Entregables: tabla de fuentes; afirmaciones con enlaces; lista de información no verificada; resumen general; web accesible sin analítica ni publicación.

Límites: fecha de corte, citas breves, separación de hechos y opinión y respeto de licencias.

Comprobación: abrir las fuentes de cinco afirmaciones y revisar la web en móvil, escritorio y teclado.

Proyecto D - iniciación al desarrollo

Resultado: ampliar mi-primer-web con una funcionalidad pequeña.

Entregables: AGENTS.md; plan; cambio acotado; prueba o procedimiento; diff y commit local; informe de límites.

Límites: sin dependencias ni publicación; conservar cambios ajenos; una idea por commit.

Comprobación: comportamiento visible, teclado, pantalla estrecha y revisión Git.

Prompt de inicio

Voy a realizar el proyecto final [A/B/C/D]. No actúes todavía. Entrevístame para completar solo las decisiones que cambian el resultado. Después prepara: objetivo, materiales autorizados, entregables, límites, plan, comprobaciones, caso de fallo, recuperación y fuentes. Separa obligatorio de opcional y espera mi aprobación.

Rúbrica

Criterio	Puntos	Evidencia
Objetivo y alcance	15	Resultado, entregables y exclusiones
Fuentes y contexto	15	Origen, fecha, límites y trazabilidad
Seguridad y privacidad	15	Datos mínimos, permisos y ausencia de secretos
Resultado útil	20	Abre, se entiende y cumple el objetivo
Verificación	20	Caso correcto, caso problemático y revisión manual
Recuperación	10	Copia/Git e instrucciones reproducibles
Comunicación final	5	Hechos, riesgos y no verificado

Superación: 75 puntos y ningún fallo crítico de seguridad, pérdida de originales o publicación no autorizada.

Entrega final

```
# Entrega del proyecto
## Objetivo
## Materiales y permisos
## Resultado
## Archivos creados o modificados
## Comprobaciones realizadas
```

```
## Caso de fallo probado
## Información no verificada
## Cómo repetir
## Cómo deshacer
## Fuentes
## Próximo paso recomendado
```

La persona revisora debe comprender el resultado, abrirlo, repetir la comprobación y conocer sus límites sin leer toda la conversación.

Cierre del curso

Has terminado cuando puedes responder: qué superficie elegiste, qué acceso concediste, qué cambió, qué evidencia demuestra que funciona, qué no se comprobó, cómo vuelves atrás y qué acción todavía necesita una persona.

Error frecuente

Elegir el proyecto más grande. Un resultado pequeño, completo y verificable demuestra más control que una idea ambiciosa sin revisión, recuperación o límites claros.

Evidencia que debes guardar

- Plan aprobado y materiales autorizados.
- Archivos de entrada y entrega.
- Caso correcto y caso problemático.
- Rúbrica cumplimentada.
- ENTREGA.md y revisión de otra persona.

Ejemplo resuelto completo: el kit de lanzamiento de Marta

Marta elige un proyecto que combina las lecciones 4, 5, 6, 7, 11, 14 y 15: una web local con materiales de lanzamiento, sin publicar ni contactar a nadie.

1. Define el resultado. Una web de una página, un inventario de materiales y una lista de pendientes. El proyecto termina cuando otra persona puede abrir la web, revisar la información y seguir las instrucciones.

2. Prepara el entorno. Crea entrada, web, salida y respaldo. Copia únicamente materiales ficticios o duplicados. Guarda un inventario inicial.

3. Encarga con límites.

```
Construye en web/ una página local para el servicio descrito en entrada/brief.md.
Usa solo los recursos de entrada/recursos. No inventes precios, testimonios,
clientes ni certificaciones. Si falta información, utiliza [PENDIENTE].
Antes de editar, presenta estructura y preguntas. Después implementa una
versión funcional para móvil y escritorio. No publiques, no envíes formularios
y no instales servicios externos. Crea salida/informe-final.md con archivos,
decisiones, pruebas ejecutadas, pendientes e instrucciones para abrir la web.
```

4. Revisa durante el proceso. Marta responde las preguntas, aprueba la estructura y corrige una interpretación del público. No espera a que toda la web esté terminada.

5. Comprueba. Compara inventario inicial y final, abre la página en dos anchos, recorre enlaces con teclado, busca marcadores [PENDIENTE], revisa el diff o mapa de cambios y pide una segunda inspección crítica.

6. Entrega. El informe indica qué se hizo, qué no se hizo y cómo reemplazar datos provisionales. Una persona ajena ejecuta las instrucciones. Si necesita ayuda oral, la documentación no está terminada.

7. Reflexiona. Marta anota: «El prompt inicial no definía si el formulario debía enviar. Convertirlo en un

elemento visual sin envío evitó una acción externa no deseada».

Variaciones de proyecto por perfil

- **Sol:** paquete de estudio con índice de fuentes, actividades y rúbrica; no publicar en el aula.
- **Leo:** dossier de investigación con tabla de afirmaciones, fuentes y fecha de consulta.
- **Dani:** mejora pequeña de un repositorio con issue, plan, diff, pruebas y documentación.
- **Equipo mixto:** una persona encarga, otra revisa evidencia y una tercera intenta seguir las instrucciones desde cero.

Rúbrica de autoevaluación del proyecto

Puntúa cada criterio de 0 a 2: objetivo y alcance; protección de originales; calidad del encargo; correcciones durante el proceso; evidencia; accesibilidad o usabilidad; recuperación; documentación final. Un proyecto con menos de 12 sobre 16 necesita otra ronda. Cualquier cero en seguridad, evidencia o recuperación impide considerarlo terminado.

Comprueba que lo has entendido

1. ¿El proyecto final se evalúa por cantidad de archivos o por evidencia de que resuelve el problema?
2. ¿Qué tres elementos no pueden faltar en el informe final?

Solución comentada

Se evalúa por el resultado y sus controles, no por volumen. Como mínimo: cambios o entregables producidos, comprobaciones realmente ejecutadas y límites o pendientes. También conviene incluir instrucciones de uso y recuperación.

Fuentes oficiales

- [Prompting](#)
- [Projects, chats, and tasks](#)
- [Agent approvals & security](#)
- [Best practices](#)

Revisado contra la documentación: 14 de julio de 2026.

Apéndice A. Matriz rápida por plataforma

Función	macOS	Windows	Linux
ChatGPT para escritorio	Sí	Sí	No
Codex local en la app	Sí	Sí	No
Codex CLI	Sí	Sí, nativo o WSL2	Sí
Extensión de IDE	Sí	Sí	Sí
ChatGPT Work en la web	Sí	Sí	Sí
Navegador integrado	App y web	App y web	Web; no CLI/IDE
Computer Use local	Sí, con permisos del sistema	Sí, escritorio activo	No como app local
Revisión Git local	Con Git instalado	Con Git instalado	Con Git instalado
Scheduled con carpeta local	App encendida	App encendida	No mediante app local
Sandbox local	Seatbelt	Windows sandbox o WSL2	bubblewrap recomendado

La disponibilidad también depende de cuenta, plan, región, despliegue y política de la organización.

Apéndice B. Plantillas de prompts

Solo lectura

Trabaja únicamente en [carpeta]. No cambies archivos ni ejecutes acciones externas. Analiza [material] y devuelve [salida]. Separa hechos de inferencias, indica qué no pudiste leer y cita rutas concretas.

Plan antes de actuar

No actúes todavía. Inspecciona el contexto, formula solo las preguntas que cambian el resultado y propón un plan breve. Para cada paso indica archivos, permisos, riesgo, comprobación y recuperación. Espera mi aprobación.

Crear sin sobrescribir

Crea el resultado dentro de [salida] con un nombre nuevo. No modifiques ni borres originales. Muestra primero una vista previa. Después vuelve a abrir el resultado, comprueba [criterios] y declara lo no verificado.

Investigación

Investiga [pregunta] con fecha de corte [fecha]. Prioriza fuentes primarias. Incluye enlace directo para cada afirmación, separa hechos de inferencias y escribe "no verificado" cuando la fuente no respalde el dato. No realices acciones externas.

Corregir el rumbo

Detén [acción]. El estado observado es [hecho] y esperaba [resultado]. Conserva [elementos], descarta [pasos] y propón la corrección más pequeña. No continúes hasta que confirme el plan.

Cierre con evidencia

Vuelve a leer los archivos, realiza las comprobaciones acordadas y entrega: cambios, evidencia, límites, riesgos y recuperación. No presentes como comprobado lo que no ejecutaste.

Borrador sin enviar

Prepara [mensaje/publicación/formulario] como borrador local. No abras cuentas, no busques destinatarios, no adjuntes y no envíes ni publiques. Marca los datos que requieren mi confirmación.

Apéndice C. Solución de problemas

No puedo iniciar sesión

Comprueba fecha y hora, dominio oficial, cuenta y espacio de trabajo. Si existe SSO, MFA o administración, consulta a la organización. No desactives MFA ni pegues códigos en el chat.

Codex no ve la carpeta

Confirma proyecto y directorio actual, revisa permisos del sistema y no abras toda tu carpeta personal para resolverlo.

Puede leer, pero no escribir

Revisa el perfil, confirma que la salida está dentro del workspace y amplía únicamente la ubicación necesaria.

Pide red repetidamente

Identifica comando y dominio. Decide si basta una fuente ya disponible o una búsqueda. Si la red no estaba en el plan, detén y revisa.

Browser, Computer Use, plugins o Scheduled no aparecen

Comprueba superficie, Plugins, Settings, plan, espacio de trabajo y política administrativa. Actualiza la aplicación y utiliza la alternativa de la lección.

Linux avisa del sandbox

Comprueba bubblewrap y la guía oficial para tu distribución. No desactives globalmente protecciones del kernel como primera solución.

Windows mezcla PowerShell y WSL

Decide dónde viven proyecto y herramientas. Usa Windows nativo o un flujo WSL2 coherente. Reinicia la app después de cambiar el agente. No uses WSL1.

Computer Use controla otra ventana

Detén, cierra aplicaciones sensibles, autoriza una sola aplicación y retira permisos persistentes si no son apropiados.

No hay evidencia

Pide reabrir archivos, ejecutar la comprobación y distinguir ejecutado, observado e inferido. Conserva «no verificado» si no puede comprobarlo.

Cambió demasiados archivos

Detén, revisa lista y diff, conserva cambios ajenos, restaura solo el alcance de la tarea y reduce el objetivo antes de reintentar.

La tarea no termina

Pide estado y bloqueo, divide investigación, creación y publicación, y no aumentes permisos o coste sin comprender la causa.

Diagnóstico técnico

`codex doctor` puede generar un informe de instalación, configuración y autenticación. Revísalo antes de compartirlo porque puede contener rutas y nombres de proyecto.

Apéndice D. Glosario

Acción externa: enviar, publicar, comprar, borrar o cambiar algo fuera de la carpeta local.

Agente: sistema que utiliza herramientas y avanza varios pasos hacia un resultado.

API key: credencial de la plataforma API con facturación y controles propios.

Aprobación: decisión sobre una acción concreta.

Chat: preguntas y conversación.

ChatGPT Work: investigación y entregables revisables.

CLI: interfaz de terminal.

Codex: agente de OpenAI para software y tareas técnicas.

Commit: versión registrada por Git.

Computer Use: operación de aplicaciones gráficas autorizadas.

Contexto: información disponible para comprender la tarea.

Diff: comparación de cambios.

Entregable: resultado final revisable.

Git: control de versiones local; no publica por sí solo.

IDE: editor preparado para proyectos de software.

Inferencia: conclusión que no aparece explícita en una fuente.

MCP: protocolo para conectar modelos con herramientas y fuentes.

Plugin: paquete instalable con skills y/o conexiones.

Prompt: pregunta, instrucción u objetivo.

Proyecto: agrupación de tareas y contexto o carpeta conectada.

Repositorio: carpeta cuyo historial gestiona normalmente Git.

Sandbox: frontera técnica de archivos, comandos y red.

Skill: flujo reutilizable con instrucciones y recursos.

Tarea: conversación orientada a un resultado.

Worktree: copia de trabajo aislada de un repositorio Git.

Workspace: carpeta o conjunto de carpetas del trabajo local.

Apéndice E. Plan de capturas para la futura edición

Las capturas se harán después de aprobar el texto y nunca sustituirán las instrucciones:

1. Selector Chat / Work / Codex.
2. Proyecto local limitado a Aprender–Codex.
3. Control de permisos.
4. Primera tarea de solo lectura.
5. Previsualización y anotación de un documento.
6. Navegador con hostname visible.
7. Ajustes de Computer Use en macOS.
8. Computer Use en primer plano en Windows.
9. Web local en escritorio y ventana estrecha.
10. Panel Git con un único fragmento.
11. AGENTS.md e instrucciones activas.
12. Plugin con permisos visibles.
13. Scheduled con una tarea de práctica.
14. CLI con /status, ocultando datos personales.
15. Rúbrica del proyecto final.

Cada imagen debe registrar sistema, versión y fecha; ocultar datos personales; conservar visible el control relevante; tener texto alternativo; y comprobarse en pantalla e impresión.

Apéndice F. Fuentes y revisión

Fuentes oficiales principales

- [Quickstart](#)
- [Use ChatGPT](#)
- [Prompting](#)
- [Best practices](#)
- [Projects, chats, and tasks](#)
- [Work with files](#)
- [Authentication](#)
- [Sandbox](#)
- [Agent approvals & security](#)
- [Browser](#)
- [Computer Use](#)
- [Scheduled tasks](#)
- [Skills & Plugins](#)
- [Model Context Protocol](#)
- [Custom instructions with AGENTS.md](#)
- [CLI reference](#)
- [IDE reference](#)
- [Windows app](#)
- [WSL2](#)

Revisión antes de publicar

1. Obtener la documentación oficial vigente.
2. Comprobar nombres de productos y superficies.
3. Probar instalación e inicio de sesión en macOS, Windows y Linux; WSL2 por separado.
4. Completar cada práctica desde una carpeta nueva.
5. Registrar versión, sistema y resultado real.
6. Comprobar enlaces y alternativas de funciones variables.
7. Hacer una revisión con una persona sin experiencia técnica.
8. Hacer revisión de accesibilidad y lectura impresa.

Datos deliberadamente no fijados

No se fijan precios, límites exactos, un modelo obligatorio ni ubicaciones de interfaz sujetas a rediseño. La versión publicada debe enlazar la fuente vigente y mostrar fecha de revisión.

Apéndice G. Paquete de práctica reproducible

Este paquete permite realizar las actividades sin utilizar información personal. Todos los nombres, correos, importes y situaciones son ficticios. Crea una carpeta llamada Aprender-Codex y reproduce esta estructura:

```
Aprender-Codex/  
├── 01-primeros-archivos/  
│   ├── ideas.txt  
│   ├── notas.txt  
│   ├── seguimiento.txt  
│   └── tareas.txt  
├── 02-datos/  
│   └── entrada/  
│       ├── clientes-con-incidencias.csv  
│       └── clientes-semana-2.csv  
├── 03-web/  
│   ├── brief.md  
│   └── recursos/  
│       └── textos.md  
├── 04-investigacion/  
│   └── preguntas.md  
└── salida/  
    └── respaldo/
```

Deja vacías salida y respaldo. Durante el curso, una práctica solo puede escribir en ellas o en la carpeta que su lección autorice expresamente.

Archivo ideas.txt

IDEAS PARA EL TALLER

- Una actividad de 15 minutos para explicar qué es una fuente primaria.
- Un ejemplo con una noticia y el comunicado original.
- Tarjetas para distinguir hecho, opinión e inferencia.
- Terminar con una lista de comprobación.
- Tal vez grabar un vídeo; todavía no está aprobado.

Archivo tareas.txt

TAREAS

Revisar objetivos - Sol - 2026-07-18 - pendiente
Preparar tarjetas - Alex - 18/07/2026 - en curso
Confirmar aula - sin responsable - 2026-07-16 - pendiente
Publicar materiales - Sol - sin fecha - bloqueada hasta revisión

Este archivo contiene fechas con formatos distintos y datos ausentes. Codex debe detectarlos, no completar lo que falta.

Archivo notas.txt

NOTAS DE LA REUNIÓN

El grupo tendrá entre 12 y 18 personas.
 La sesión durará 90 minutos, aunque el borrador anterior decía 60.
 No utilizar cuentas personales durante la demostración.
 Preparar alternativa impresa por si falla la conexión.
 Pendiente: confirmar si el aula tiene proyector.

Archivo seguimiento.txt

SEGUIMIENTO

Objetivos: revisados.
 Tarjetas: primera versión lista; falta comprobar lectura fácil.
 Aula: reservada, pero el proyector no está confirmado.
 Vídeo: no iniciado y no aprobado.
 Publicación: no autorizada.

Los cuatro archivos permiten practicar inventario, contradicciones y planificación. Por ejemplo, la duración de 90 minutos debe prevalecer como dato más reciente solo si el contexto permite justificarlo; si no, se marca para revisión.

Archivo clientes-con-incidencias.csv

```
cliente,email,ultima_contacto,estado
Alba Norte,alba@example.test,2026-06-28,pendiente
Bosque Azul,bosque@example.test,2026-07-03,en curso
Casa Río,,2026-07-05,pendiente
Delta Taller,delta@example.test,05/07/2026,pendiente
Estudio Faro,faro@example.test,fecha-desconocida,cerrado
Ficción Sur,ficcion@example.test,2026-07-09,en curso
```

Los dominios .test están destinados a ejemplos y no deben utilizarse para envíos. La fecha 05/07/2026 es interpretable en España, pero no cumple el formato declarado; fecha-desconocida es inválida; Casa Río no tiene correo. El resultado correcto conserva las seis filas y señala las incidencias.

Archivo clientes-semana-2.csv

```
cliente,email,ultima_contacto,estado
Alba Norte,alba@example.test,2026-07-11,en curso
Bosque Azul,bosque@example.test,2026-07-10,cerrado
Casa Río,,2026-07-05,pendiente
Grupo Luna,luna@example.test,2026-07-12,pendiente
```

Este segundo archivo sirve para comprobar si la automatización es reutilizable. No se debe combinar con el primero salvo que el encargo explique cómo resolver clientes repetidos.

Archivo brief.md

Brief: Taller Norte

Taller Norte ofrece sesiones de organización digital para pequeños comercios.

Público

Personas que utilizan correo, documentos y hojas de cálculo, pero no se consideran técnicas.

Objetivo de la web

Explicar el servicio y facilitar que una persona prepare una consulta. La web de práctica será local: no enviará formularios ni se publicará.

Mensajes confirmados

- Sesiones prácticas con ejemplos cotidianos.
- Material de seguimiento después de cada sesión.
- Adaptación al nivel del grupo.

Datos pendientes

- Precio.
- Duración exacta.
- Dirección de contacto.
- Testimonios: no existen y no se deben inventar.

Archivo recursos/textos.md

Textos provisionales

Título: Organización digital sin complicaciones

Descripción: Aprende un método claro para ordenar archivos, revisar tareas y reducir trabajo repetitivo.

Llamada a la acción provisional: Prepara tu consulta

Nota: cualquier correo o precio debe mostrarse como [PENDIENTE].

Archivo preguntas.md

Investigación de práctica

Compara tres fuentes públicas sobre alfabetización mediática.

1. ¿Cómo define cada fuente una fuente primaria?
2. ¿Qué método propone para verificar una afirmación?
3. ¿Qué fecha y ámbito geográfico tiene la información?
4. ¿Qué coincide y qué discrepa?

No redactes una recomendación hasta completar una tabla de evidencia.

Control del paquete antes de empezar

Comprueba que existen 9 archivos: cuatro de texto iniciales, dos CSV, un brief, un recurso de texto y un archivo de preguntas. Guarda una copia comprimida o duplica la carpeta en `respaldo/inicio`. Si una práctica modifica entrada, recupera el paquete antes de continuar.

Apéndice H. Cuatro casos integrales

Los casos integrales no introducen funciones nuevas. Combinan varias lecciones y obligan a tomar decisiones. Intenta cada caso antes de leer los criterios de corrección.

Caso 1. Sol prepara una unidad didáctica

Situación. Sol tiene apuntes, una lista de fuentes y resultados de una actividad. Necesita un paquete para una sesión de 90 minutos, pero no quiere publicar ni utilizar datos del alumnado.

Entregables. `guia-docente.md`, `actividad-alumnado.md`, `rubrica.md` y `control-de-fuentes.md`.

Secuencia recomendada.

1. Duplicar materiales y retirar nombres o identificadores.
2. Pedir inventario sin cambios y marcar contradicciones.
3. Definir objetivos observables: identificar fuente primaria, distinguir hecho e inferencia y justificar una verificación.
4. Solicitar un esquema antes de redactar.
5. Generar los cuatro archivos en una salida nueva.
6. Revisar nivel de lectura, tiempo real de la actividad, accesibilidad y correspondencia entre objetivos y rúbrica.

Prompt de partida.

Usa únicamente los materiales anonimizados de entrada. Diseña una sesión de 90 minutos para principiantes. Antes de redactar, muestra contradicciones, preguntas y distribución del tiempo. No inventes fuentes ni experiencias del alumnado. Cada afirmación factual debe poder rastrearse a `control-de-fuentes.md`. Guarda los cuatro entregables indicados en salida y termina con una revisión de accesibilidad, tiempos y datos pendientes. No publiques nada.

Trampa deliberada. Una nota antigua indica 60 minutos. El agente debe advertir la discrepancia y no mezclar ambas duraciones.

Criterios de corrección. Los tiempos suman 90 minutos; la rúbrica evalúa los objetivos declarados; no hay datos personales; todas las fuentes están identificadas; los archivos pueden usarse sin una explicación oral adicional.

Caso 2. Marta prepara el seguimiento semanal

Situación. Cada semana llega un CSV parecido, pero puede contener columnas o formatos diferentes. Marta necesita saber a quién revisar; durante el curso no se enviará ningún mensaje.

Entregables. `seguimiento.csv`, `incidencias.csv`, `resumen.md` y `registro-ejecucion.md`.

Secuencia recomendada.

1. Validar cabeceras, codificación y número de filas.
2. Mostrar cinco filas de vista previa.
3. Separar datos procesables e incidencias sin perder filas.

4. Reconciliar totales.
5. Repetir con la semana 2 en una carpeta de salida distinta.
6. Comparar resultados sin fusionar registros por intuición.

Trampa deliberada. Casa Río carece de correo y dos fechas no cumplen el formato. Un resultado que complete esos datos está mal aunque parezca útil.

Criterios de corrección. Las seis filas de la primera semana aparecen exactamente una vez entre seguimiento e incidencias; cada regla está documentada; no se modifica la entrada; la segunda ejecución no sobrescribe la primera; no se producen envíos.

Caso 3. Leo redacta una comparativa con fuentes

Situación. Leo compara tres opciones para una asociación. Debe separar información oficial, opiniones de terceros e inferencias propias.

Entregables. protocolo.md, tabla-evidencia.md, comparativa.md y registro-fuentes.md.

Secuencia recomendada.

1. Escribir las preguntas y criterios antes de buscar.
2. Priorizar fuentes primarias y registrar la fecha de consulta.
3. Anotar afirmaciones, evidencia, vigencia y confianza.
4. Detenerse ante contradicciones importantes.
5. Redactar la comparativa desde la tabla, no desde memoria.
6. Hacer una auditoría inversa de cinco afirmaciones.

Prompt de auditoría.

Elige las cinco afirmaciones más importantes de comparativa.md. Para cada una, localiza la fila exacta de tabla-evidencia.md y la fuente correspondiente. Marca SIN RESPALDO cualquier frase cuyo camino no sea completo. No busques fuentes nuevas en esta fase: evalúa únicamente lo entregado.

Trampa deliberada. Una reseña secundaria es más reciente que la página oficial, pero no explica de dónde obtiene un dato. La fecha reciente no la convierte automáticamente en autoridad.

Criterios de corrección. Hechos e inferencias se distinguen; las fechas tienen significado claro; las discrepancias no se ocultan; ninguna conclusión importante queda sin respaldo.

Caso 4. Dani corrige una incidencia pequeña

Situación. Una web permite enviar un formulario con un correo claramente inválido. El repositorio ya contiene un cambio local ajeno a la incidencia.

Entregables. Cambio mínimo, prueba reproducible y informe-incidencia.md.

Secuencia recomendada.

1. Leer instrucciones y consultar el estado de Git.
2. Identificar y proteger el cambio preexistente.
3. Reproducir el problema antes de editar.

4. Proponer un plan breve y archivos afectados.
5. Implementar la corrección y una prueba relevante.
6. Revisar diff, ejecutar pruebas y realizar comprobación manual.

Trampa deliberada. Una herramienta de formato intenta cambiar todo el archivo. Dani debe separar ese ruido de la corrección, no aceptar un diff enorme por comodidad.

Criterios de corrección. El fallo se demuestra antes y después; el cambio ajeno permanece intacto; el diff se limita a la validación; el informe distingue pruebas ejecutadas de comprobaciones pendientes; no se publica ni crea un commit sin autorización.

Cómo realizar los casos en grupo

Asigna tres papeles rotatorios: **persona solicitante**, que define el resultado; **persona operadora**, que dirige a Codex; y **persona revisora**, que solo puede utilizar la evidencia entregada. Si la revisora necesita creer una afirmación no demostrada, la entrega vuelve a la persona operadora. Cambiad los papeles en el siguiente caso.

Apéndice I. Clínica de prompts

Un prompt se mejora tomando decisiones, no añadiendo palabras decorativas. La columna central no es la única formulación correcta: muestra el tipo de precisión que falta.

Petición débil	Petición mejorada	Qué se hizo explícito
«Ordena esto.»	«Sin modificar originales, inventaría los archivos, propón nombres y muestra colisiones antes de copiar a salida.»	Alcance, secuencia y protección.
«Hazlo profesional.»	«Usa tono formal, títulos descriptivos, fechas uniformes y una tabla legible en pantalla e impresión.»	Sustituye un adjetivo por criterios.
«Investiga a fondo.»	«Responde cuatro preguntas, prioriza fuentes oficiales, registra fecha y separa hechos de inferencias.»	Preguntas, fuentes y trazabilidad.
«Arregla la web.»	«Reproduce el desbordamiento a 360 px, corrígelo sin cambiar el contenido y comprueba que no haya desplazamiento horizontal.»	Fallo, alcance y prueba.
«Mándaselo al cliente.»	«Prepara un borrador dirigido a la persona indicada; no lo envíes. Muestra destinatario, asunto y adjuntos para revisión.»	Separa preparación y acción externa.
«Limpia el CSV.»	«Conserva todas las filas, no inventes ausencias, separa incidencias y reconcilia recuentos.»	Define qué significa limpiar.
«Usa fuentes fiables.»	«Prioriza documentación primaria; justifica las secundarias y marca cualquier afirmación sin respaldo directo.»	Convierte fiabilidad en protocolo.
«Mejora el código.»	«Reduce la duplicación de la función indicada sin cambiar su comportamiento; ejecuta las pruebas y muestra el diff.»	Localiza, limita y verifica.
«Automatiza esto cada semana.»	«Primero ejecútalo manualmente; luego programa una salida fechada, registra fallos y no envíes ni sobrescribas.»	Prerrequisito, evidencia y fallos.
«Haz una copia.»	«Copia la carpeta en respaldo/antes-del-cambio, verifica número de archivos y no borres copias anteriores.»	Destino y comprobación.
«Comprueba que funciona.»	«Ejecuta la prueba indicada, abre el resultado, informa del comando y separa lo comprobado de lo pendiente.»	Evidencia concreta.
«Termínalo.»	«Compara el resultado con estos cinco criterios, corrige solo incumplimientos y entrega cambios, pruebas, pendientes e instrucciones.»	Define el cierre.

Ejercicio 1. Diagnostica antes de reescribir

Para cada petición débil, identifica qué falta en estas cinco categorías:

1. **Resultado:** ¿qué debe existir al final?
2. **Contexto:** ¿qué datos, público o situación importan?
3. **Alcance:** ¿qué puede leer o cambiar?
4. **Límites:** ¿qué no debe hacer ni inventar?
5. **Prueba:** ¿cómo sabrás que está bien?

No todos los encargos necesitan las cinco categorías con el mismo detalle. Una explicación sencilla puede requerir solo resultado y nivel. Una acción sobre archivos o cuentas suele necesitarlas todas.

Ejercicio 2. Reduce sin perder control

Este prompt es claro pero repetitivo:

No modifiques entrada. No cambies entrada. Entrada debe permanecer igual.
Haz una copia en salida. Quiero que el resultado esté en salida. No lo pongas en entrada. Comprueba todo y asegúrate de que está bien y correcto.

Una versión más útil sería:

Trata `entrada/` como solo lectura y escribe únicamente en `salida/`.
Al terminar, compara recuentos, abre una muestra e informa de cualquier error.

La reducción conserva la regla y concreta «comprueba todo» en acciones observables.

Ejercicio 3. Añade una pausa inteligente

Transforma una tarea de alto impacto para que se detenga antes del último paso:

Prepara la publicación con título, texto, imagen y audiencia. No publiques.
Detente en la pantalla final y muéstrame un resumen para confirmación.

La pausa no compensa un contexto equivocado, pero evita convertir una revisión en una acción irreversible.

Plantilla universal mínima

Quiero [resultado].
Usa [contexto o archivos] y trabaja solo en [alcance].
No [límites o acciones excluidas].
Antes de actuar, [plan, preguntas o vista previa].
Al terminar, comprueba [criterios] y entrega [evidencia].

Si una línea no aporta una decisión real, elimínala. Si falta una decisión que cambiaría el resultado, añádela.

Apéndice J. Autoevaluación y guía para facilitar el curso

Autoevaluación inicial

Antes de empezar, puntúa de 0 a 2 cada afirmación: 0, todavía no; 1, con ayuda; 2, sin ayuda.

1. Puedo distinguir una conversación, un entregable visual y una tarea sobre archivos.
2. Sé localizar la carpeta con la que estoy trabajando.
3. Puedo explicar qué permiso se me solicita.
4. Sé describir un resultado y dos criterios de aceptación.
5. Puedo conservar originales y separar salidas.
6. Sé pedir evidencia en vez de aceptar «hecho».
7. Puedo detener una acción externa antes de confirmarla.
8. Sé recuperar una práctica desde una copia o Git.

La puntuación no determina si puedes hacer el curso. Sirve para comparar el inicio con el final. Entre 0 y 5, empieza por la ruta esencial con acompañamiento; entre 6 y 11, sigue la ruta general; entre 12 y 16, puedes acelerar las primeras prácticas, pero debes demostrar igualmente sus controles.

Evaluación final por demostración

No preguntes solo definiciones. Entrega a la persona una carpeta de práctica con una contradicción, una ausencia y un archivo que no debe modificarse. Pídele que:

1. inspeccione sin cambios;
2. formule preguntas;
3. proponga un plan;
4. ejecute una salida limitada;
5. compruebe recuentos y una muestra;
6. explique qué sabe, qué infiere y qué queda pendiente;
7. recupere el estado inicial.

Se considera competente cuando realiza el ciclo sin perder el original, puede justificar permisos y no confunde una respuesta persuasiva con una comprobación.

Rúbrica común de 0 a 3

Dimensión	0	1	2	3
Objetivo	No existe	Es vago	Es claro	Es claro y medible
Alcance	Sin límites	Límites implícitos	Carpeta y acciones definidas	Incluye exclusiones y casos dudosos

Dimensión	0	1	2	3
Seguridad	Arriesga originales	Depende de una advertencia	Conserva originales	Añade vista previa y recuperación probada
Observación	Acepta sin mirar	Mira el resultado final	Revisa plan y cambios	Corrige el rumbo con criterio
Evidencia	Solo afirma	Evidencia parcial	Controles relevantes	Controles reproducibles y límites explícitos
Explicación	No puede explicarlo	Repite la respuesta	Explica decisiones	Explica decisiones, incertidumbre y recuperación

Una puntuación alta no exige lenguaje técnico. Exige control y comprensión.

Guía para una persona facilitadora

Antes de la sesión: prepara el paquete desde cero, comprueba disponibilidad de las superficies, utiliza cuentas y archivos de demostración, decide qué acciones estarán prohibidas y prueba la alternativa sin conexión.

Durante la sesión: proyecta también los errores, verbaliza por qué se aprueba o rechaza un permiso, deja tiempo para leer el plan y pide predicciones antes de ejecutar. No rescates inmediatamente a quien se equivoca: pregunta «¿qué evidencia esperabas?» y «¿cómo volverías atrás?».

Después de la sesión: conserva las entregas de práctica sin datos personales, registra dudas recurrentes y actualiza las capturas o pasos sujetos a cambios de producto. Revisa la documentación oficial antes de cada nueva edición.

Adaptaciones de accesibilidad

- Proporciona los prompts también como texto copiable, no solo en capturas.
- Lee en voz alta las solicitudes de permiso y explica el efecto.
- Evita depender únicamente del color para verde, ámbar y rojo; incluye las etiquetas.
- Divide las prácticas largas en puntos de parada con resultado visible.
- Permite utilizar teclado, lector de pantalla o ampliación durante la demostración.
- Ofrece los archivos con nombres consistentes y una versión comprimida limpia.
- No evalúes velocidad de escritura; evalúa decisiones, evidencia y recuperación.

Preguntas para cerrar cada módulo

1. ¿Qué decisión nueva sabes tomar?
2. ¿Qué error eres capaz de detectar ahora?
3. ¿Qué evidencia guardarías para otra persona?
4. ¿Qué acción todavía no autorizarías sin ayuda?
5. ¿Cómo recuperarías el estado inicial?

Si las respuestas son abstractas, repite una práctica con un archivo distinto. El aprendizaje se consolida cuando el método sobrevive al cambio de ejemplo.

Lista final de aprobación editorial

- ☐ El título y la promesa corresponden a un curso general desde cero.
- ☐ Chat, Work y Codex están diferenciados sin menospreciar ninguna superficie.
- ☐ macOS, Windows y Linux tienen rutas explícitas.
- ☐ Las primeras seis lecciones no exigen Git.
- ☐ Las primeras cinco lecciones en macOS/Windows no exigen terminal.
- ☐ Cada práctica utiliza entrada, salida, verificación y recuperación propias.
- ☐ Las acciones externas permanecen como borrador durante el aprendizaje.
- ☐ Los originales se conservan.
- ☐ Cada función de producto tiene fuente oficial específica.
- ☐ Cada lección incluye un ejemplo resuelto y una autoevaluación razonada.
- ☐ El paquete de práctica se puede reconstruir literalmente y no contiene datos reales.
- ☐ Los cuatro casos integrales producen evidencia revisable.
- ☐ La clínica de prompts enseña a concretar, no solo a alargar.
- ☐ La rúbrica puede ser utilizada por una persona sin experiencia técnica.
- ☐ «Revisado» y «probado» no se confunden.
- ☐ Las capturas futuras tendrán texto alternativo y no contendrán datos personales.
- ☐ El proyecto final puede completarse desde un perfil no técnico.
- ☐ La ruta técnica enlaza el curso existente sin duplicarlo.
- ☐ No se inicia maquetación PDF/LaTeX hasta recibir aprobación.