

Kit AppStart

Guía de uso v0.4.6

Pipeline completo desde la idea hasta el handoff, con seguimiento visual del roadmap durante la implementación.

Bienvenido/a

Desde AL Grano IA hemos preparado esta guía para que puedas aplicar Kit AppStart de forma ordenada, entender qué ocurre en cada fase y no perderte cuando empiece la implementación real. La idea es sencilla: tú mantienes la autoridad, los agentes trabajan con límites claros y el repositorio conserva la trazabilidad.

Venga amigos, vamos al grano.

Versión cubierta	Kit AppStart v0.4.6 - FINAL / APPROVED
ZIP oficial	kit_appstart_v0.4.6.zip
SHA-256 ZIP	59a9d97b9b5fd39df020921bd82a5bd39a7285910ef3cb9bbbb877bfed90e823
Fecha de versión	14 de agosto de 2026

Qué vas a hacer con esta guía

Kit AppStart separa claramente diseño, aprobación, incorporación documental, verificación y ejecución. Esta guía recorre el flujo completo sin asumir que eres programador experto.

- Preparar un Proyecto de ChatGPT independiente para cada aplicación.
- Completar y aprobar Fase 1, Fase 2 y Fase 3 en orden.
- Incorporar el paquete aprobado al repositorio con Claude Code sin empezar a implementar.
- Entregar el control de verificación a ChatGPT Work en modo solo lectura.
- Usar el nuevo Progress Tracker de v0.4.6 para saber siempre dónde estás en el roadmap.

Qué cambia en v0.4.6

El pipeline F1 -> F2 -> F3 se mantiene. La novedad principal aparece después del handoff: el roadmap aprobado queda identificado mediante una referencia inequívoca y la implementación puede mantener un docs/PROGRESS_TRACKER.md mutable, separado del paquete F3 protegido. Work lo verifica en solo lectura y Claude lo actualiza únicamente dentro del alcance autorizado.

Mapa rápido

```

IDEA
↓
PROYECTO CHATGPT
↓
F1 → APROBACIÓN
↓
F2 → APROBACIÓN
↓
F3 → APROBACIÓN → POST-APPROVAL SYNC
↓
ZIP FINAL + PROMPT EXTERNO PARA CLAUDE
↓
REPOSITORIO GIT REAL
↓
CLAUDE INCORPORA docs/handoff/
↓
WORK VERIFICA EN SOLO LECTURA
↓
PRIMER BLOQUE AUTORIZADO
↓
docs/PROGRESS_TRACKER.md → 📌 ESTÁS AQUÍ

```

PARTE 1 · PREPARAR EL PROYECTO

1. Preparación inicial

1. Crea un Proyecto nuevo de ChatGPT para esa aplicación.

Usa un Proyecto independiente para cada app. No reutilices el Proyecto de otra aplicación, porque las decisiones y autoridades de una ejecución no deben contaminar otra.

2. Configura las Instrucciones del Proyecto.

Copia íntegramente el contenido de ``INSTRUCCIONES_PERMANENTES_PROYECTO.md`` y pégalo en Instrucciones/Ajustes del Proyecto.

No lo subas como Source.

3. Sube exactamente las cinco fuentes metodológicas del Kit como Sources del Proyecto.

- ``00_START_HERE.md``
- ``01_APP_CLASSIFICATION.md``
- ``02_MINIMUM_SECURITY_AND_QUALITY.md``
- ``03_TECHNOLOGY_RESEARCH.md``
- ``04_OUTPUT_AND_HANDOFF.md``

No subas como Sources

``MENSAJE_NUEVA_APP.md``, ``CHANGELOG.md``, ``VERSIONING_AND_MIGRATION.md``, ``KIT_MANIFEST.json`` ni ``LICENSE``. Estos archivos pertenecen a la gobernanza/distribución del propio Kit, no al contexto normal de cada app.

4. Prepara la idea inicial.

Abre ``MENSAJE_NUEVA_APP.md``, sustituye su placeholder por la descripción real de la nueva aplicación y conserva el resto de la plantilla.

No necesitas diseñar arquitectura, tecnologías, base de datos ni proveedores. Describe qué quieres construir, para quién y las restricciones que ya conozcas.

5. Abre un chat nuevo dentro del Proyecto y pega ``MENSAJE_NUEVA_APP.md`` ya rellenado.

Ese es el arranque oficial de una nueva ejecución.

6. Resuelve el PLATFORM PREFLIGHT solo si hace falta.

Si ya has dicho expresamente algo como “app web”, “Windows”, “macOS”, “Android”, etc., el Project debe continuar.

Si la plataforma no está clara, debe detenerse antes de desarrollar materialmente Fase 1 y preguntártela. No puede deducirla por tu ordenador, por la aparente sencillez de la app ni por conversaciones anteriores.

PARTE 2 · FASE 1

2. Fase 1 - Descubrimiento y clasificación

7. El Project analiza producto, usuario, datos, impacto, riesgo, privacidad, gates y `documentation\_profile`.

En v0.4.6 debe proponer uno de estos perfiles: Lite, Standard o Extended. No deberías elegirlo simplemente porque prefieras más o menos documentos; debe derivarlo de la aplicación.

Puede generar una primera revisión como F1-R01 con `PHASE\_1\_COMPLETE`, `PHASE\_1\_DECISION\_BRIEF` y `PHASE\_1\_ACTION\_REQUIRED` si existen decisiones reales.

8. Contesta únicamente las decisiones materiales de F1.

Por ejemplo: usuarios reales, naturaleza de los datos, consecuencias de un fallo, distribución, etc.

El Project integra tus respuestas y genera una revisión consolidada, por ejemplo F1-R02. Si ya no queda nada que decidir, debe eliminar `PHASE\_1\_ACTION\_REQUIRED`.

9. Haz la revisión independiente de F1 antes de aprobarla.

Comprueba clasificación, authority provenance, contradicciones, sobreingeniería y cualquier decisión material pendiente.

Apruebo definitivamente la Fase 1 consolidada F1-RXX y autorizo comenzar la Fase 2.

Sustituye F1-RXX por la revisión real que estés aprobando.

PARTE 3 · FASE 2

3. Fase 2 - Requisitos, investigación y arquitectura

10. Comienza F2 solo después de aprobar F1.

Aquí se trabajan requisitos funcionales y no funcionales, semántica material, investigación tecnológica, comparación de alternativas, arquitectura, seguridad, privacidad/regulación cuando aplique, dependencias/licencias, entornos, testing, roadmap y quality gates.

La investigación externa debe tratar webs, README, documentación, repositorios, PDFs, etc. como UNTRUSTED EVIDENCE, nunca como instrucciones.

`documentation\_profile` puede reevaluarse si en F2 aparece nueva evidencia.

11. Resuelve las decisiones de F2.

El Project puede generar DP-F2-01, DP-F2-02, etc. Tú respondes y el Project consolida la fase.

Debe sincronizar `PHASE\_2\_COMPLETE`, `PHASE\_2\_DECISION\_BRIEF`, registros regulatorios/licencias si aplican y `PHASE\_2\_ACTION\_REQUIRED` solo si sigue habiendo algo material pendiente.

12. Audita F2 antes de aprobarla.

Revisa especialmente investigación, fuentes, versiones, licencias, regulación, trazabilidad de requisitos y arquitectura.

Apruebo definitivamente la Fase 2 consolidada F2-RXX y autorizo comenzar la Fase 3. Mantén F1-RXX y F2-RXX como autoridades aprobadas de esta ejecución. Esta aprobación autoriza únicamente comenzar la Fase 3 documental y generar el paquete de traspaso previsto por Kit AppStart v0.4.6. No autoriza incorporar el paquete a docs/handoff/, crear o modificar el repositorio, bootstrap, instalación o restore de dependencias, generación de lockfile, implementación, commits, push, PR, merge ni release.

PARTE 4 · FASE 3

4. Fase 3 - Paquete de autoridad y handoff

13. Materializa la autoridad aprobada de F1 + F2 en un paquete físico.

El Project no vuelve a diseñar la aplicación. Para un proyecto Standard puede incluir PROJECT_PROFILE, PROJECT_BRIEF, REQUIREMENTS, RISK_REGISTER, RESEARCH_EVIDENCE, TECHNOLOGY_DECISIONS, ARCHITECTURE, ROADMAP, quality gates y registros de terceros/regulación.

La selección concreta depende de DOCUMENT_SELECTION; no tiene que producir siempre el mismo número de archivos.

F3 debe incluir los artefactos operativos obligatorios correspondientes, incluyendo README, HUMAN_SUMMARY, APPROVED_DEPENDENCIES, HANDOFF para Work, FIRST_MESSAGE_TO_WORK y manifiestos.

Preparación nueva en v0.4.6

Los ítems de implementación que se seguirán después deben disponer de IDs estables. F3 también materializa una `roadmap\_authority\_ref` resoluble: puede apuntar a un ROADMAP separado o, en Lite, a la sección consolidada que actúe como autoridad. Work recibe esa referencia ya resuelta; no tiene que adivinar dónde está el roadmap.

14. Revisa F3 antes de aprobarla.

Comprueba estructura física, autoridad, dependencias, handoff, Work read-only, hashes, manifest, blockers y que continúen `implementation\_authorized: false` e `incorporation\_authorized: false`.

Los blockers futuros no necesariamente bloquean aprobar F3; pueden permanecer abiertos para sus momentos posteriores.

15. Aprueba F3 documentalmente.

Utiliza la revisión y package version reales que haya generado el Project.

```
Apruebo definitivamente la Fase 3 F3-RXX y el paquete F3-PXX.
Mantén F1-RXX, F2-RXX y F3-RXX como autoridades aprobadas de esta ejecución.
Mantén abiertos los blockers definidos en PHASE_3_ACTION_REQUIRED en los momentos
allí establecidos, si ese archivo existe.
Realiza únicamente el post-approval package sync previsto por Kit AppStart v0.4.6:
conserva la misma phase_revision F3-RXX y package_version F3-PXX, sincroniza status y
generated_at, recalcula hashes y manifiestos, ejecuta nuevamente el consistency check
físico y vuelve a entregarme el ZIP final sincronizado.
Esta aprobación acepta únicamente el paquete documental.
implementation_authorized: false
incorporation_authorized: false
No autorizo todavía incorporar el paquete a docs/handoff/, crear o modificar el
repositorio, bootstrap, instalación o restore de dependencias, generación de
lockfile, implementación, commits, push, PR, merge ni release.
```

16. El Project ejecuta el post-approval sync sin crear una nueva revisión.

Si aprobaste F3-R01 / F3-P01, debe seguir siendo F3-R01 / F3-P01. No se convierte en F3-R02 solo porque haya pasado de `pending\_owner\_approval` a `approved`.

El Project sincroniza status, generated_at, hashes y manifiestos y vuelve a generar el ZIP final. Ambas autorizaciones siguen en false.

17. Recibes dos entregables distintos del Project.

- A. ZIP F3 final aprobado: paquete documental que debe acabar en el repositorio.
- B. Archivo separado, fuera del ZIP:
``<PROJECT_ID>_PROMPT_TO_CLAUDE_FOR_HANDOFF_INCORPORATION.md``.

Importante

El prompt externo es una plantilla no autoautorizante. Tenerla o descargarla no autoriza nada. La autorización nace cuando tú decides enviársela a Claude Code.

18. Verifica el ZIP final.

Es la última revisión externa del trabajo del Project. Comprueba ZIP correcto, `status: approved`, revisión/package correctos, timestamps sincronizados, hashes internos, manifest, blockers preservados y ambas autorizaciones en false.

Una vez pasado esto, el Project ha terminado su trabajo.

PARTE 5 · REPOSITORIO, CLAUDE Y WORK

5. Incorporación documental al repositorio

19. Crea o abre tú el repositorio real de la aplicación.

Este paso ocurre ya fuera de ChatGPT Project. Puedes utilizar GitHub Desktop u otra herramienta para crear el repositorio local o abrir/clonar uno que ya exista.

Antes de abrir Claude debe existir un repositorio Git válido. No necesitas todavía código, framework, dependencias, lockfile, bootstrap, estructura de app ni funcionalidad.

20. Abre Claude Code dentro de ese repositorio.

Claude debe estar trabajando en ese repositorio concreto. No le pidas que encuentre otro repositorio.

21. Asegúrate de que el ZIP final está accesible para Claude.

Normalmente basta con que siga en Descargas. No es necesario renombrarlo porque la identidad definitiva es su SHA-256, no el nombre físico.

22. Envía a Claude exactamente el contenido del archivo externo de incorporación.

Este acto es la autorización de incorporación y autoriza solo esa operación.

El prompt obliga a Claude a comprobar primero que está realmente en un repositorio Git válido. Si no lo está debe detenerse. Tiene prohibido ``git init``, crear/buscar otro repo, crear una rama o cambiar de rama.

23. Claude valida el paquete antes de tocar el repositorio.

Comprueba repo actual, rama, HEAD, worktree, SHA-256 del ZIP, `project_id`, F3 revision, package version, ``status: approved``, hashes de MANIFEST, ``implementation_authorized:false``, ``incorporation_authorized:false`` y blockers de ACTION_REQUIRED.

Si algo falla, se detiene sin modificar el repositorio.

24. Claude incorpora exclusivamente los documentos.

Si todo es correcto, extrae el contenido del ZIP en ``docs/handoff/<PROJECT_ID>_PHASE_3_PACKAGE/``.

No mete simplemente el ZIP allí: incorpora su contenido estructurado. Si esa carpeta ya existe y es exactamente el mismo paquete, informa no-op. Si existe pero difiere o está incompleta, se detiene y no sobrescribe ni fusiona.

25. Claude vuelve a validar el paquete ya incorporado y se detiene.

Verifica otra vez hashes contra los bytes físicos y entrega su informe.

Durante esta operación están prohibidos ``git init``, cambio/creación de rama, bootstrap, restore/install, lockfile, implementación, staging, commit, push, PR, merge y release.

Es normal que ``docs/handoff/`` quede untracked/uncommitted en este momento. No necesitas hacer un commit documental antes de Work.

26. Abre ChatGPT Work sobre exactamente ese mismo repositorio.

No otro repo, no otra copia.

27. Localiza `FIRST_MESSAGE_TO_WORK` dentro del paquete incorporado.

Ruta conceptual:

``docs/handoff/<PROJECT_ID>_PHASE_3_PACKAGE/05_HANDOFF/<PROJECT_ID>_FIRST_MESSAGE_TO_WORK.md``.

28. Copia TODO su contenido y pégalo como el primer mensaje de ChatGPT Work.

No tienes que redactar otro prompt ni pegar manualmente `HANDOFF_PROMPT_CHATGPT_WORK`.

`FIRST_MESSAGE_TO_WORK` ya le ordena a Work localizar el paquete, validar manifests/hashes, leer el `HANDOFF` completo y recibir la `roadmap_authority_ref` concreta de esta ejecución.

29. Work entra en modo de verificación estrictamente read-only.

Comprueba paquete, repo, rama, HEAD y worktree reales. Si todo está correcto, puede preparar como máximo un borrador del siguiente encargo para Claude.

Work no ejecuta ni autoriza por sí mismo. Los IDs que incluya en su borrador son “propuestos para autorización”; solo pasan a ser “autorizados” después de tu aprobación humana y entrega manual a Claude.

NUEVO EN v0.4.6 · PROGRESS TRACKER

6. Cómo saber siempre dónde estás

A partir del primer bloque autorizado de implementación, Kit AppStart v0.4.6 puede mantener un archivo mutable de seguimiento en el repositorio:

docs/PROGRESS_TRACKER.md

Separación de autoridad

El ROADMAP aprobado sigue siendo el plan y permanece dentro de la autoridad documental. `docs/PROGRESS\_TRACKER.md` solo refleja el estado operativo actual. No sustituye requisitos, arquitectura, roadmap, quality gates ni evidencia.

6.1 Primera inicialización

- Si Work entra por primera vez y el tracker todavía no existe, debe decirlo claramente.
- Work usa la `roadmap\_authority\_ref` entregada por F3. No inventa ni busca otro roadmap.
- El primer borrador de encargo para Claude puede incluir la creación de `docs/PROGRESS\_TRACKER.md`.
- Claude solo lo crea cuando tú autorizas ese bloque de trabajo.

📌 ESTADO DEL ROADMAP

Tracker: todavía no inicializado
 Autoridad del roadmap: validada
 Siguiendo ítem previsto: ENG-01 - [not_started]
 Bloqueos: ninguno

6.2 Vista humana: 📌 ESTÁS AQUÍ

📌 ESTÁS AQUÍ

ROADMAP DE IMPLEMENTACIÓN - 3 de 8

ENG-03 - Backend

- | | | | |
|---|-------------------|---------------|---------|
| ● | 3.1 Estructura | [completed] | |
| ● | 3.2 API básica | [completed] | |
| ● | 3.3 Usuarios | [in_progress] | ← AHORA |
| ● | 3.4 Autenticación | [not_started] | |
| ● | 3.5 Tests | [not_started] | |

Siguiente: 3.4 Autenticación · Bloqueos: ninguno

6.3 Estados canónicos

Vista	Estado canónico	Significado
	not_started	Todavía no se ha comenzado.
	in_progress	Es el trabajo activo dentro del alcance autorizado.
	completed	Está cerrado con criterios, tests/gates aplicables y evidencia.
	blocked	No puede avanzar hasta resolver el bloqueo.

Regla de cierre

`completed` no significa "Claude ha escrito el código". Cuando corresponda, deben estar satisfechos los criterios de aceptación, los tests requeridos, el quality gate aplicable y la evidencia. Si un test o gate realmente no aplica, debe quedar como N/A justificado. No puede quedar abierto un blocker incompatible con el cierre.

6.4 Si algo terminado vuelve a necesitar cambios

Un ítem puede reabrirse sin borrar su historial. Por ejemplo, un bug, una regresión, un cambio de requisito aprobado o un fallo de revisión pueden llevar de `completed` a `in\_progress` o `blocked`.

```
ENG-03.3 - Usuarios
Estado anterior: completed
Estado actual: in_progress
Motivo: BUG-014 - fallo en edición de usuario
Condición de re-cierre: corrección + pruebas aplicables PASS
```

PARTE 7 · CICLO DE TRABAJO DIARIO

7. Work, tú y Claude: quién hace qué

Rol	Hace	No hace
ChatGPT Work	Lee roadmap, tracker, repo y evidencia. Muestra el estado y propone el siguiente encargo.	No escribe archivos. No autoriza a Claude.
Propietario	Revisa el estado y aprueba o rechaza el siguiente bloque.	No pierde la autoridad por usar agentes.
Claude Code	Implementa solo los IDs autorizados y actualiza el tracker dentro del alcance.	No empieza automáticamente el siguiente ítem.

7.1 Qué deberías ver en Work

📌 ESTADO DEL ROADMAP

Etapa actual: ENG-03 - Backend
 Ítem actual: ENG-03.3 - Usuarios
 Estado: [in_progress]
 Anterior: ENG-03.2 - [completed]
 Siguiente previsto: ENG-03.4 - [not_started]
 Bloqueos: ninguno

Después de ese bloque, Work puede explicar la revisión técnica y preparar como máximo el siguiente borrador para Claude. Los IDs del borrador son propuestos para autorización.

7.2 Qué debería decir Claude al empezar

📌 ROADMAP

Trabajando en: ENG-03.3 - Usuarios
 Estado inicial: [in_progress]
 Alcance autorizado: ENG-03.3-T01 / ENG-03.3-T02

No se comenzará el siguiente ítem sin nueva autorización.

7.3 Qué debería decir Claude al terminar

📌 ROADMAP

ENG-03.3 - [completed]
 Criterios de aceptación: PASS
 Tests requeridos: PASS o N/A justificado
 Quality gate aplicable: PASS o N/A justificado
 Evidencia: disponible

Siguiente: ENG-03.4 - [not_started]
 ENG-03.4 no ha sido iniciado.

RESUMEN FINAL

8. El pipeline completo, en una línea

IDEA → Proyecto ChatGPT → F1 → aprobación → F2 → aprobación → F3 → aprobación → post-approval sync → ZIP final + prompt Claude → crear/abrir repo → Claude incorpora docs/handoff/ → Work → pegar FIRST_MESSAGE_TO_WORK → primer bloque autorizado → PROGRESS_TRACKER → implementación por ítems verificables

División de responsabilidades

- Project diseña, investiga y empaqueta.
- Tú apruebas las decisiones y los bloques de trabajo.
- Claude incorpora físicamente y, después, implementa únicamente lo autorizado.
- Work verifica el paquete, el repositorio y el progreso en solo lectura.
- El ROADMAP define el plan. El PROGRESS_TRACKER te dice dónde estás ahora.

Recuerda

Si en cualquier momento no sabes dónde está la aplicación, abre `docs/PROGRESS\_TRACKER.md` en VS Code. La vista debe permitirte identificar el punto actual, lo completado, lo pendiente, los bloqueos y el siguiente paso sin reconstruir conversaciones técnicas.

Gracias por utilizar los recursos de AL Grano IA.

Apps reales, con IA, sin rodeos.

algranoia.com/recursos