MARCO MOTION

PROTOCOL

WORKING PROTOCOL WITH AI AGENTS

A standardized technical process for running several AI coding agents (Claude Code) in parallel: per-agent isolation, a record at the close of every session, context-cost control, backup and security. It includes the web layer that organizes, edits, publishes and exports the documents the process produces.

Marco Francisco · Updated 2026-10-08

1. ARCHITECTURE: ONE AGENT PER DIRECTORY

- **Unit of work:** one directory is one agent, with its own startup context (`CLAUDE.md`) and its own persistent memory. No agent reads another agent's memory.
- **Two instruction levels:** global memory (the user's rules for every agent) and per-project memory.
- **Per-project memory:** one fact per file, with metadata (`name`, `description`, `type`) and four types: user, feedback, project and reference. A one-line-per-entry index is the only part loaded on every request; topic files are opened on demand.
- **Memory is bound to the directory the session is opened from, not to the repository:** opening the same project from two paths produces two independent indexes.

2. ARTIFACTS RECORDED PER SESSION

ARTIFACT	PURPOSE	FORMAT
Startup context (`CLAUDE.md`)	What the agent reads on start	Only what causes a failure if unknown: rules, deployment traps, closed decisions. Capped at 13–14 KB; reference material moves to `docs/` and leaves a pointer
Changelog	Chronological index	One short entry per session; no detail
Session file	Full record	Summary · files modified · files created · decisions · deploys · tests · lessons · open items
Decision record (ADR / D-NNN)	Traceability	Context, decision, author, date and which earlier decision it repeals
Professional log	Evidence-backed professional profile	In English. Per session: *Direction & Decision-Making* · *Technical Strategy* · *Creative* · *Areas of Opportunity* · *Cumulative Profile Update*. Records decision patterns, not lists of deliverables. Source for the skills map and the CV
Development tree	Inventory of real state per branch	States, one line per feature (see below)
Performance buffs	The agent's operating manual toward the user	Behavior → amplification → effect (see below)
Memory	Facts that persist across sessions	Absolute dates; an existing file is updated before a duplicate is created

Development tree. Allowed states:

STATE	MEANING
`[x]`	shipped and verified in production
`[~]`	shipped, end-to-end test pending
`[!]`	blocked, open bug or anomaly
`[ ]`	not started

STATE	MEANING
`[-]`	deferred or retired, with the reason on the line

Rules: one line per feature, 60 characters at most, with the session suffix `(Sxx)`; two heading levels at most; a feature that spans branches is listed in each. The close applies incremental deltas; the tree is never rebuilt.

Performance buffs. Each entry follows the sequence: user behavior → how the agent amplifies it → effect on the user's cognitive load. Categories: pacing, communication, execution, verification and debugging; an anti-patterns section; a last-reviewed line. If the session revealed no new pattern, the file is left untouched. It does not duplicate the professional log: that one observes for the record; this one states what the agent must do.

One source per document. Each document exists as a single markdown file; the web version and the PDF are generated from it. An automated test compares the SHA-256 fingerprint of the source with the one recorded at the last generation and blocks the deploy if they differ. It removes drift between versions of the same document. This protocol is published that way.

3. STANDARDIZED SESSION CLOSE

- **One local skill per agent**, with a unique name (`cierre-<agent>`) and never the name of a global skill: when names collide, the personal skill takes precedence over the project one. Invoked from another agent's session, the name does not resolve, which works as a guard.
- **Common skeleton, fixed order:**

STEP	CONTENT
0	Smoke gate: paths tested, with evidence. Blocks the documentation
1	Changelog
1.x	Development tree
2	Session file
2.x	Professional log, performance buffs and decisions
3	`CLAUDE.md` : the history keeps only the latest session; over the cap, the close warns and does not trim
4	Memory
5	Commit by path; push per the repository's policy; without git, the backup state is reported
6	Report with evidence

- **Each agent's own steps** are derived from evidence in the repository (its `CLAUDE.md` , its latest session files and the records of its previous closes), not from a template. Each step cites its source.
- **How the closes are produced:** one read-only subagent per directory drafts in parallel; a mechanical check covers name, size, banned terms and sections; installation is a commit limited to that path.

4. CONTEXT AUDIT AND TOKEN-COST OPTIMIZATION

- **Cost model:** every request resends the full context (instructions, `CLAUDE.md`, memory index and conversation history). A session's cost is the sum of the context of all its requests.
- **Measurement:** from the sessions' JSONL logs, deduplicated by `requestId` (streaming leaves ~1.7 entries per request). Metrics: floor, average, maximum, total and number of compactions (`compact_boundary`).
- **Targets:** `CLAUDE.md` ≤ 13 KB; memory index ≤ 17 KB and ≤ 135 lines. Above ~24 KB the index is truncated when read: paid in full, not read in full.
- **Trim by relocation, never by deletion:** reference sections move to `docs/` with a pointer; a line-by-line check then confirms that all original content exists in some destination and that no reference is left orphaned.
- **Compaction window:** the model alias was found to have moved to a 1M-token window, with automatic compaction near 967K. Sessions of hundreds of requests were measured with zero compactions and peaks of 900–965K. Countermeasure: an explicit cap (`autoCompactWindow`), validated by counting compactions in the log. Estimate with a 300K cap: average context per request drops from ~500K to ~200K (≈2.5×).
- **Operational hygiene:** read ranges, not whole files; subagents return conclusions, not content; the tool set (MCP) is not changed mid-session because it invalidates the prompt cache (one measured invalidation: 332,903 tokens); coordination between sessions by messaging was discarded on cost.
- **Measured proportion:** trimming `CLAUDE.md` accounted for ~1% of the spend; the dominant factor was the uncompacted window.

5. WEB PLATFORM FOR DOCUMENT ORGANIZATION

- **Structure:** document repository by project → section → document, with a viewer.
- **Source of truth: the disk.** One-way publishing with a distribution manifest and change detection by SHA-256. A guard compares the local copy against the remote before publishing, so an outdated copy cannot overwrite what is published, and aborts if it cannot read the published state.
- **Four publishing modes:** text (markdown to database), file (self-contained HTML to object storage), binary (PDF or image, untransformed) and folder.
- **In-browser editing:** edits are saved as drafts; publishing is a separate privilege. Documents can be created from the browser. Publishing commits automatically to the private documents repository.
- **Server-side PDF generation** through an API-driven headless browser.
- **Share links** with access control and expiry; printable as a QR code.
- **Site content:** compiled text as the floor and an editable patch on top; each section's visibility is data (a flag), not a code branch.
- **Operational modules:** calendar of events and open calls (with source, text and notes), contacts directory, and outgoing mail with blind copy and a sent log.
- **Documentation map** generated from counts read from disk, with a verifier that fails when the map no longer matches the files.

6. Interactive editing and automation of professional documents

- **Editor with a live preview of the real PDF** next to the full document, with a page counter.
- **Symbolic deletion:** text is struck through, not removed; each mark follows its text even when the text moves.
- **Named variants:** create, duplicate, publish and delete; one published per language; entries and bullets reordered by drag and drop.
- **Layout engine:** primitives shared by the web renderer and the PDF renderer; blocks that measure themselves and return a stroke plan; region-based fitting that never truncates silently; a schema with mixed tokens (millimeters and fractions), regions, page papers and ornaments, validated per layout.
- **Per-document settings:** destination, layout, palette, target and colors; insertable cover pages.
- **PDF with real hyperlinks** (annotations). The public download honors the layout of the published variant.
- **Generators from markdown:** reference document with cover, presentation, and PDF printed by a headless browser with reduced motion; three visual dialects derived from the same design tokens.
- **Automated tests** that read the text of the generated PDF (ToUnicode map decoding).

7. Backup and security

Backup

- Private repositories. Content (md, html, scripts) is versioned in git; multimedia stays out through `.gitignore`.
- Derived and binary files in private storage with no public domain, with the inventory kept inside the storage itself; restoring verifies file by file against its hash.
- A backup counts only once verified: files on the remote = versioned files.

Security

- Secrets and credentials kept out of the repository; one key per product, so revoking one does not affect the others.
- Writes to another project's production database only with the user's explicit authorization in the chat.
- In repositories shared between agents, each commit includes only the agent's own paths.
- **The organization of security is not described in this document: publishing it represents a vulnerability.**

8. Failures detected: continuous learning

> *«Se vale fallar, eso me ayudó a perfeccionar este sistema». — Marco Francisco*

DETECTED	COUNTERMEASURE
A single close template applied to different repositories: nonexistent steps marked N/A or run falsely; 5 of 11 agents with an unclosed session	Per-agent close, with steps derived from its own evidence
Documentation drift: a capabilities document 52 sessions behind the canonical manual	Only the manual regenerated at every close is cited
Loss of qualifiers when compressing a rule («hasta verificarse», *until verified*, was dropped)	Check against the original text when trimming
A configuration assumption invalidated by a model change (alias with a 1M window)	Validation by measurement, not by declared configuration
Zero-coverage checks; recursive searches that skip files ignored by git or detected as binary	Every check reports how many items it examined; searches force text reading
3 of 11 directories without external backup	Private git, content only
Memory bound to the directory of origin: two indexes for the same project	The audit covers every memory directory