

SIX-WEEK PROGRAMME FOR DEVELOPMENT TEAMS

AI productivity programme

Get more work out of AI, spend fewer tokens and be able to show it

Six weeks with Claude Code and GitHub Copilot: I measure the starting point, we run the workshop, I review the team's pull requests and I measure again. At the end the team has a shared method and two measurements that can be compared.

6

weeks, with two measurements

5 to 8 h

of on-site workshop, depending on the team

2

tools: Claude Code and GitHub Copilot

5

weeks of remote follow-up

WHERE TEAMS START

The team uses AI every day, each one in their own way, and nobody knows if it pays off

One person uses the most expensive model to rename a variable. Another has had a session open since Monday that resends its whole history with every message. Another leaves the agent looping all afternoon. And pull requests arrive green because the same agent that wrote the code decided it was fine.

The token bill keeps growing and everyone feels faster, but nobody has measured anything. Claude Code's cost documentation gives an average of about 13 dollars per developer per active day, and the first step is knowing where the team stands.

That's why I don't promise a number. I measure the before and the after, with indicators we pick together on the call, and whatever changes, or doesn't, gets written down.

What the team takes away in six weeks

01 More work finished

The expensive model only where real thinking is needed, and parallel agents with briefs that don't collide.

02 Fewer tokens

A cache that doesn't break, context on demand and sessions that finish on their own.

03 Confidence in production

The AI tests what it generates and another agent tries to break it before the merge.

04 A shared method

The same configuration, the same agents and the same rules across every team.

05 Two comparable measurements

The same indicators at the start and at the end, to see what has changed.

We work through everything in Claude Code and in GitHub Copilot. The workshop in week 1 is the AI productivity course, in full; everything else is what surrounds it so the change can be seen.

WHAT THE PROGRAMME DELIVERS

What the team has after six weeks

Each result can be checked against something that can be shown to management, and not against a hunch.

01

More work finished with the same team

A model per round, agents working in parallel on disjoint briefs and sessions with turn and spending limits that finish on their own.

Evidence: the same task with one model for everything and with a model per round, with the cost of each.

02

Fewer tokens per finished task

A cache that doesn't break, context loaded on demand and output filtered before it reaches the model.

Evidence: tokens per task measured at the start and at the end, with the same method.

03

Confidence in what goes to production

The AI tests its own code: a test that fails first, protected tests and a suite that runs before anything is declared finished. A reviewer in a clean context looks for what the author considered fine.

Evidence: written comments on two or three of the team's agent-made pull requests.

04

One way of working across teams

Instructions, permissions, skills, hooks and agents set up in the company's repositories and agreed by the team.

Evidence: configurations on a branch of the company's repository, reviewed by the team.

05

Five prioritised actions for management

A three-to-five-page report with what I saw, what we worked on and what to do next, with one owner per action.

Evidence: the report, delivered in week 2.

What happens each week and who does it

I do almost all the work. From the team I ask for time for the survey, the workshop day and passing me the pull requests.

Week 0

Diagnosis and first measurement

The team answers a 10-minute survey about tools, habits and frustrations. I look at the team's instruction files and permissions, and together we work out the baseline for the indicators we choose.

Deliverable: A written baseline and labs adapted to the team's stack.

Week 1

The workshop

Between five and eight hours in person on a model per round, cache, context on demand, finite execution, orchestration and verification, with the last block on each participant's own repository. I leave the configurations on a branch of the company's repository.

Deliverable: Configurations on a branch and a labs repository.

Week 2

Management report

I prepare the three-to-five-page report with what I saw in the diagnosis, what we worked on, risks in the configurations (without auditing anything) and five prioritised actions.

Deliverable: Management report.

Weeks 1 to 5

Q&A channel and pull request review

A channel with the team, open until the end of week 5, for configuration and usage questions. I review in writing two or three pull requests that the team points me to.

Deliverable: Written answers and comments on the pull requests.

Week 6

Review and second measurement

We repeat the week 0 measurements with the same method and spend 90 minutes remotely looking at what has changed, what has stalled and what is still missing.

Deliverable: A before-and-after comparison and an agreed next step.

Week 1 is the full workshop. Its syllabus, labs and agenda are at danielalfocea.com/en/formacion/workshop-productividad-con-ia

What gets measured, and why I promise nothing

We pick four to six indicators on the call, based on what can be extracted from the company's own systems without building anything new. These are candidates and the list is not closed. Six weeks don't give a statistical conclusion, and the report says so.

Tokens per finished task

What each accepted change costs, taken from /usage or the tool's dashboard. It is the cheapest indicator to measure and the one that moves most when context is put in order.

Time from opening to merging a pull request

It comes from the repository history. It can't tell who wrote what, but it does show whether the flow stalls in review.

Changes reverted or fixed afterwards

How many merged changes are reverted or need a fix in the following days. It shows whether what reaches production holds up.

Review effort

Rounds of comments and diff size on agent-made pull requests. A huge diff that someone approves without reading is a signal.

Team perception

The same short survey at the start and at the end. The key question is how much time they think they save, and that can be set against what was measured.

Workshop or programme

NEED	WORKSHOP	PROGRAMME
The team sharing one way of working with agents	The workshop does it: one day, set up in the company's repositories.	It works too, and adds five weeks of follow-up.
Knowing whether anything measurable has changed	The workshop teaches how to measure a session, but there is no second measurement.	This is what the programme is for: two comparable measurements.
Reporting to management	Each participant leaves with their before and after numbers.	Adds the three-to-five-page report with five prioritised actions.
Correcting course on the team's real code	Only in the last block of the day.	I review two or three of the team's pull requests in the following weeks.

The topics behind those benefits

The full content is that of the workshop in week 1. Each topic is explained in Claude Code and in GitHub Copilot, and in the following weeks we apply it to the team's pull requests.

01 Saving: model, cache and context

Fewer tokens without losing quality.

- Model and effort per round: planning, exploring, executing and reviewing don't call for the same thing.
 - What breaks the cache, rewinding versus compacting and output filtered with hooks.
 - Context on demand, without RAG: an index file, path-scoped rules and documentation the agent walks on its own.
-

02 Finite execution and orchestration

An agent with no turn or spending limit gets expensive.

- Turn and spending limits, stop conditions and the Ralph loop with an iteration cap.
 - A worktree per agent, an expert panel with disjoint roles, a judge and an adversarial agent.
 - When not to build a multi-agent system: Anthropic measured 90% better performance in exchange for about fifteen times more tokens.
-

03 Verification

A change is finished when a check confirms it.

- A test that fails first, tests protected against agent edits and a stop gate with the suite.
 - A reviewer in a clean context and cross-validation between different models.
 - A hook stops what an instruction can't guarantee.
-

04 Configuration and tools

Each mechanism solves a different problem.

- Plan mode, skills and commands, hooks, MCP and permissions: when to use each.
- Managed configuration for the organisation and an in-house catalogue of skills and agents.
- A single configuration folder read by both Claude Code and Copilot.

MATERIAL THAT STAYS WITH THE COMPANY

What the company keeps when the programme ends

Everything is made to live in the company's repository and documents, and not in a training folder.

Baseline and second measurement

Both sets of indicators and the method to repeat them.

Management report

Three to five pages, with five prioritised actions and an owner for each.

Configurations on a branch

Instructions, permissions, hooks and subagents for Claude Code and Copilot, ready to review and merge.

Finite-execution recipes

The Ralph loop with its progress file, turn and spending limits and stop conditions.

Labs repository with solutions

Licence for internal use for one year.

Comments on the team's pull requests

In writing, with what gets accepted and why.

WHAT ELSE IS INCLUDED

Three pieces that come with the workshop

They are the same ones the standalone workshop includes. In the programme they are delivered within the six-week calendar.

+1

Pre-workshop team survey

Ten minutes per person to adapt the labs to the team's stack and tools.

+2

Configurations on a branch of the company's repository

Instruction files, permissions, hooks and subagents with their model and turn limit, ready to review and merge.

+3

Labs repository with solutions

Exercises, starter files and solutions, with a licence for internal use for one year.

WHO IT'S FOR

For teams that want to know whether AI pays off

A strong fit for

- Teams that already use Claude Code or GitHub Copilot every day and have no data on what effect it has.
- Tech leads and engineering managers who have to explain to management what the licences and tokens have bought.
- Platform teams that want shared configuration and agents across several teams.
- Teams that prefer to change one thing at a time and check the effect before moving on.

Not the right programme if

- The company wants a single day of training with no follow-up. That is what the workshop is for.
- What is expected is a guaranteed saving figure. I don't give one and I would be wary of anyone who does.
- What is needed is an audit of the company's code or the team's configurations.
- The team doesn't yet use a coding assistant on a real repository.

WHO RUNS IT

From cybersecurity at BBVA to AI at Santander



I'm Daniel Alfocea, and I've spent twenty years in development and security. At Banco Santander I was Principal Engineer in the AI department when the area was being set up. At BBVA I was Cybersecurity Tech Lead. At 42Crunch, in the UK, I was responsible for one of its API security products.

I've presented research at RSA Conference and OWASP AppSec EU, and I'm a regular speaker at RootedCON. Spain's National Cybersecurity Institute (INCIBE) has asked me to judge its competitions several times. I founded Navaja Negra and the OWASP Madrid chapter. I'm the author of widely used security tools such as GoLismero, Plecost and Dockerscan, and of LAPIS, research published on arXiv on how to describe an API to a model with fewer tokens. With Alfonso Muñoz I wrote *Secure MCP: A practical guide for developers, software architects and tech leads*.

This programme comes from how I work myself: I measure my own sessions, I split work across agents with their own model, turn limit and report, and I don't accept a change without a test I have seen fail.

GUARANTEES

What I fix without charging more

The configurations work

If the configurations I leave in the company's repository don't work, I run a free remote session to fix them.

One hour if the team gets stuck

If after 30 days the team has got stuck applying something from the course, I spend an hour with the team remotely, free.

This is not an audit and I don't promise saving percentages. I commit to measuring the same way both times and to reporting whatever comes out as it is.

Questions worth settling now

How much does it cost?

It depends on the length and the size of the team. We go through it in the first conversation and I send a fixed proposal.

How much will the team improve?

I don't know and I won't make up a number. What I deliver is a comparison of the team itself, done the same way at the start and at the end.

Are six weeks enough to see anything?

For some indicators yes, such as tokens per task or diff size. For others, such as reverted changes, there may not be enough data and the report will say so. I'd rather write "can't conclude" than an inflated conclusion.

What data do you need from us?

The survey answers, read access to the repository history for the indicators, and the instruction files. I don't need the full source code and everything is done on the company's laptops and accounts.

What if the team has no time for the follow-up?

The follow-up asks for little: answer a survey, point me to two or three pull requests and book 90 minutes at the end. If something doesn't arrive on time, I carry on with what there is and write it down.

Can it be funded through FUNDAE?

When a date is requested, I send the information the training department or the advisers need.

FORMAT

Six weeks, one team and two measurements

The programme includes the diagnosis, the full workshop, the report, the follow-up and the final review. We settle the length of the workshop and the group size on the first call. If only the day is needed, the workshop will do.

- Participants from the same team or from teams that work together.
- On-site workshop in English or Spanish, and follow-up remotely.
- Room, screen, wifi and laptops provided by the company.
- If we have to travel outside Madrid, we agree it before the date is confirmed.

Write to me at

me@danielalfocea.com

On the call we see what can be measured, how many people take part and which tools the team uses. With that I send a fixed proposal.

Can be subsidised through FUNDAE (Spanish training credit).

EMAIL

me@danielalfocea.com

PROGRAMME ONLINE

danielalfocea.com/en/formacion/programa-productividad-con-ia