

*Ser*finanza^{Banco}

Case Study

Banco Serfinanza Cuts Finding Management Time by Around 90% and Automates Its Secure Development Controls with AI

About Banco Serfinanza

Banco Serfinanza S.A. is a Colombian financial institution headquartered in Barranquilla, specializing in consumer banking, retail financing and financial solutions for small and medium-sized businesses.

As part of its growth and digital transformation strategy, the Bank has built its own technology capabilities to speed up the launch of digital products and services. **Its software development rests on two pillars: an in-house software factory and a group of specialized strategic partners who also write code for the Bank.**

The Challenge

When development speeds up and is split between an internal team and several external partners, security control can no longer rely on manual reviews. Every delivery is a chance for a code vulnerability or an insecure infrastructure configuration to reach production unnoticed.

And detection was not the problem. With periodic reviews, the security team received volumes of findings that had to be classified by hand, one by one, and then assigned. **That classification work, not the analysis, was the bottleneck.** The Bank identified three needs:

- **Continuous, consistent visibility** into the risk of all code entering the Bank, from the internal factory or a partner.
- **Automatic prioritization, not manual triage.**
- **Control inside the development cycle**, on every commit and pull request, not in periodic reviews.

The underlying goal: strengthen secure development governance with **automated, measurable evidence.**

”

Our challenge was not detecting vulnerabilities, it was doing it at the speed at which our internal development team and our external partners move.

Leonell Santiago Sanandres

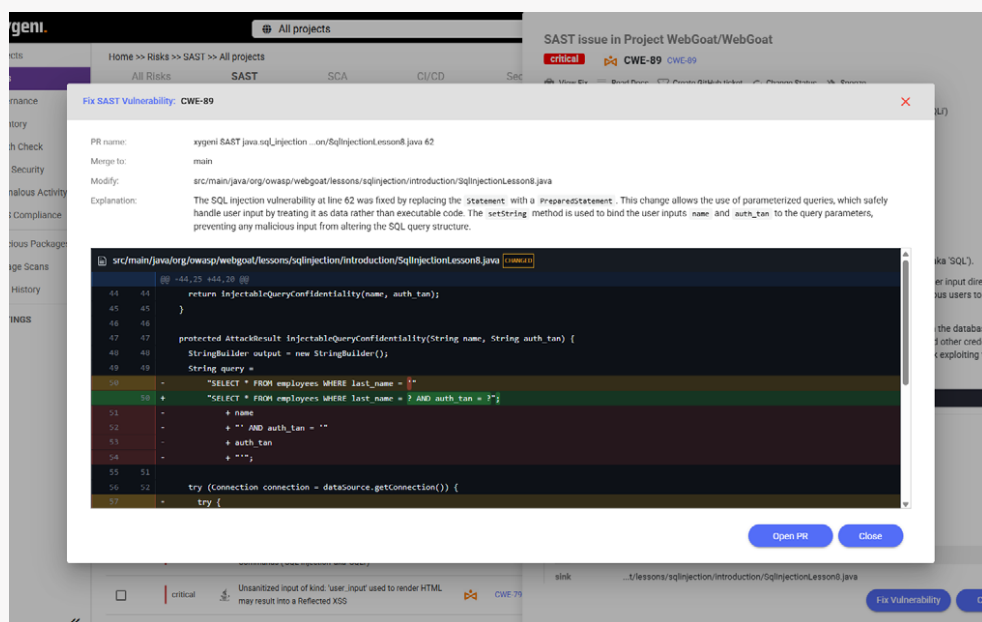
Director of Operational Risk, Banco Serfinanza

The Solution

The Bank adopted Xygeni after a thorough evaluation of the alternatives on the market. The first contact took place at **18ENISE**, the event organized by INCIBE in León, Spain. A proof of concept then validated two things: **that the platform integrated with the Bank's existing processes and tools, and that its results were actionable without adding workload to the security team.**

Xygeni covers the full cycle from a single platform: proprietary code, open source dependencies, running applications, APIs, exposed credentials and infrastructure as code, plus software supply chain protection and an ASPM layer that consolidates its own findings and those from other tools into one risk model.

Banco Serfinanza deployed four capabilities, chosen for their fit with its priorities, **integrated on GitLab** inside the existing pipelines.



Repositories were onboarded **progressively, in phases and by priority**, instead of switching on analysis across the whole estate at once. That avoided the usual effect of a security rollout: an initial volume of findings so large that development teams ignore it.

Xygeni Products Used

- **AI SAST** on proprietary code, on every commit and pull request. It uses repository context to separate the exploitable from the theoretical, across **more than 100 repositories** from the internal factory and partners alike. When a finding is confirmed, **AI AutoFix** proposes the fix in the pull request.
- **CI/CD Security** analyzes pipeline configuration as code: insecure settings, excessive permissions and build risk. With distributed development, the pipeline is where everything entering the Bank converges.
- **IaC Security** detects insecure infrastructure configurations before they are provisioned, with guardrails in the pull request, when a fix costs minutes rather than a maintenance window.
- **AI triage and prioritization** turns the volume of findings from all three into a short, ordered list, assessing whether each one is actually exploitable in that repository.

The Results

The jump in results coincides with the AI layer. Continuous detection was already solved; **what was missing was eliminating the manual work that piled up after each analysis.**

~90%

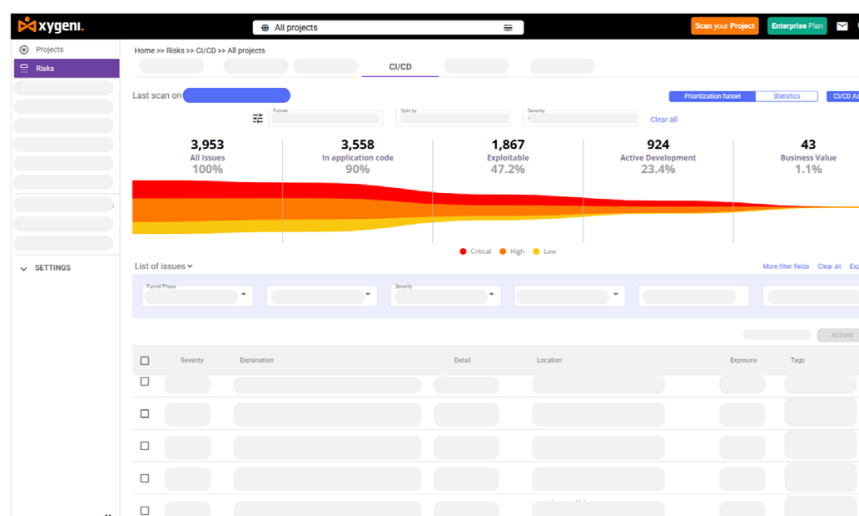
less time identifying, prioritizing
and assigning security findings

From manual triage to a short, ordered list

The savings come not from detecting faster, but from **no longer classifying by hand**. Analysis now runs on every commit and pull request, and the AI layer delivers risk ordered by real context instead of theoretical severity. Reviewing lists and deciding what to assign to whom now arrives already resolved.

Governance over code the Bank's own team does not write

The Bank moved from trusting each supplier's practices to **requiring the same verifiable standard, with automated evidence on every delivery**, across more than 100 repositories. A standard checked on every delivery does not depend on each partner's maturity, and gives the Bank a different position in conversations with its suppliers.



Early detection, before production

Code vulnerabilities and insecure infrastructure configurations are identified inside the development cycle, where fixing them costs a fraction. **Today, most findings are resolved before they reach production.**

Hours, not days, from finding to fix

A weakness now appears within minutes, **in the same pull request where it was introduced**, and is fixed while the context is fresh.

“

Implementing Xygeni has allowed us to significantly strengthen our secure development processes by automating controls and detecting vulnerabilities early. We have cut the time we spent managing and prioritizing findings by around 90%, and today most of them are resolved before reaching production. Beyond the platform's technical capabilities, we value its team's commitment to continuous improvement, innovation and close support throughout the whole process. Xygeni has become an important enabler within our digital transformation and technology risk management strategy.

Leonell Santiago Sanandres

Director of Operational Risk, Banco Serfinanza S.A.

Still triaging findings by hand?

See how Xygeni turns them into a short, ordered list.

[Book a demo](#)

[Try it free](#)