

expoQA[®]26

MADRID 26th, 27th & 28th May

expoqa.eu

LA REVOLUCIÓN MULTI-AGENTE QUE ESTÁ REDEFINIENDO EL TESTING

Los agentes ya están aquí, es un cambio de paradigma y los usamos ya en el IDE, en el pipeline, en el commit, y ha llegado al mundo del software y por lo tanto a QA.

Y como todo fuego, queman si no se aprenden a usar y no se tiene en cuenta sus ventajas y sus desafíos.



SOBRE LOS PONENTES



**JOSE
BARRAGÁN**

Senior Enterprise Architect con más de 20 años de experiencia en IT. Está especializado en tecnologías empresariales e inteligencia artificial aplicada al SDLC para impulsar modelos más eficientes y escalables.

Actualmente lidera el equipo de desarrollo y arquitectura de soluciones en ALTEN Spain.



**JAIME
RODRÍGUEZ**

Perfil técnico centrado en aplicación de IA en el ciclo de desarrollo de software incluyendo QA, automatización de pruebas, DevOps y Seguridad.

En ALTEN, lidera iniciativas de inteligencia artificial aplicada desde el equipo de arquitectura de soluciones.

CONTENIDO

ExpoQA 26

1. Agentes IA en el SDLC
2. De ChatGPT a los agentes IA
3. Rol del QA
4. Retos del paradigma agéntico
5. ALTEN AI Framework
6. Caso real en directo
7. Próximos pasos

AGENTES IA EN EL SDLC

Durante años automatizamos pasos; ahora los agentes nos dan como automatizan el razonamiento detrás de cada decisión de desarrollo, Testing o DevOps supervisada por un humano.



ANTES → AHORA



De los pasos al razonamiento

Antes automatizábamos secuencias. Ahora delegamos decisiones complejas a sistemas que las justifican.

VELOCIDAD × CALIDAD



Calidad sin renunciar a velocidad

El software se entrega más rápido y la exigencia de calidad sube en paralelo. No es uno u otro.

EL ROL CLAVE



QA es la palanca, no el cuello de botella

Quien define los criterios, supervisa el razonamiento y valida la salida sostiene todo el ciclo.

DE CHATGPT A LOS AGENTES IA

De los primeros LLMs a los sistemas multi-agente
hay cuatro saltos en cuatro años.



AGENTES IA Y QA

El Rol de QA en la generación de código será el de garantizar que se cumple con los criterios de calidad, seguridad y mantenibilidad antes de su integración.



APORTAR CONTEXTO QA

No solo negocio y desarrollo

CRITERIOS

- Specs y criterios de aceptación
- Convenciones de equipo
- Restricciones no funcionales

Contexto



VALIDAR EL OUTPUT

Revisión, no solo lo que probábamos.

CRITERIOS

- Cobertura efectiva, no aparente
- Regresión sobre cambios del agente
- Falsos positivos y código muerto

Validación



AÑADIR IA AL CICLO

Usar IA para las tareas de QA

CRITERIOS

- Generación de casos y datos
- Triage y priorización de defectos
- Análisis de impacto y selección

Aceleración

RETOS DEL PARADIGMA AGÉNTICO

Los principales desafíos del desarrollo nos obligan a resolver desde el control del contexto hasta la seguridad.



Control de contexto

El contexto se gestiona, no se asume.

Documentación, arquitectura y convenciones de equipo formalizadas.



Expectativas vs realidad

Los agentes no son perfectos.

Son multiplicadores del esfuerzo experto, no sustitutos.



Madurez técnica

Adopción gradual, no big-bang.

Requiere inversión en formación y en infraestructura de soporte.



Revisión humana

La supervisión experta es imprescindible.

Para validar outputs y evitar falsos positivos en producción.



Validación real

No basta con que compile.

Los tests generados deben verificarse contra comportamiento real.



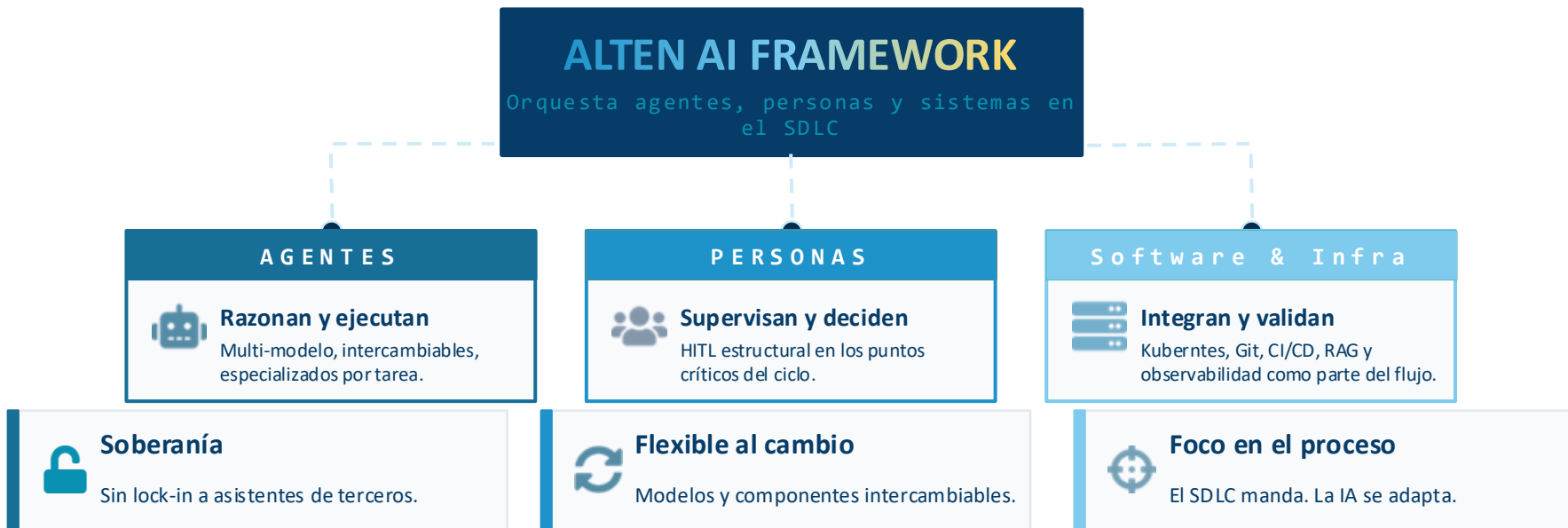
Seguridad

Nuevos vectores, mismos riesgos.

Aislamiento de credenciales, prompt injection y control de coste.

ALTEN AI FRAMEWORK

Nexus no es una herramienta. Es un framework para el desarrollo de software que no depende de terceros.



*Más allá de los asistentes. **Más allá de los agentes.** Un marco propio para llevar la IA al SDLC con criterio.*

ALTEN FRAMEWORK

Madurez técnica

Un stack por capas donde cada pieza es intercambiable: UI propia, agentes orquestados, infraestructura de soporte.

POR QUÉ ALTEN AI FRAMEWORK

Humano en el loop

HITL antes de generar código.

API-first

Integración vía REST con cualquier tool.

Skills en BD

Cambia comportamiento sin tocar código.

LiteLLM abstraer el LLM

Cambio de modelo por configuración.

Git + CI/CD

Rama verificada y probada automáticamente.

Namespaces aislados

Fallos contenidos, escala independiente.

Modelos como infra

Intercambiables desde configuración.

ALTEN-ui

React · HITL · Dashboard
checkpoint humano · tiempo real

ALTEN-api

FastAPI · JWT · REST
tareas · proyectos · estado

Capa agéntica

LangGraph · Skills · LangChain · PostgreSQL · RAG
Orquestador · Arquitecto · Implementador · Juez · RAG

LiteLLM Proxy Platform

gateway unificado · rate limits · abstracción de modelo
costes · callbacks a evaluación y observabilidad

CI/CD

Jenkins · ArgoCD · GitOps

Rancher · K8s

RKE2 · Helm · HPA

Modelos LLM

Mistral · GPT-4 · Azure

ALLEN AI FRAMEWORK

Control de contexto

Los agentes son el cerebro del sistema; el contexto, su mapa y su brújula. Lo preparan los humanos (skills, datos, código y pipelines CI/CD) y es lo que permite a los agentes generar con criterio y producir salidas de calidad.



UI

Interfaz: configuración, contexto y selección.

API /
AGENTES

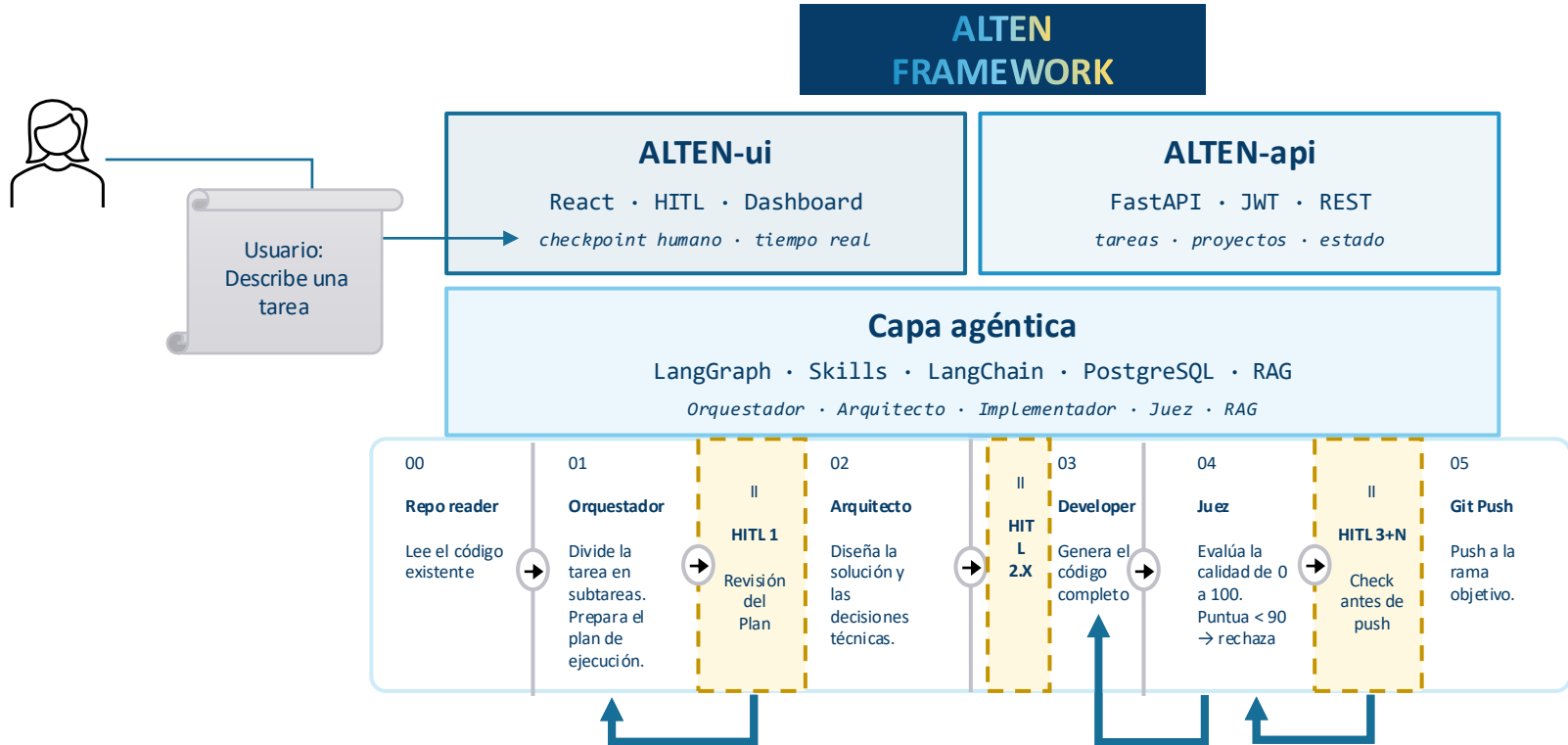
Ejecución: repo, RAG y entrega trazable.

QA

ALTEN AI FRAMEWORK

Revisión humana

Los agentes de nuestro framework son motivados por tareas: su pipeline de creación de código lo inicia una persona asignando una tarea a los agentes y cada salida debe validarse antes de subir el código.



ALTEN AI FRAMEWORK

Seguridad, expectativas y validación real

La trazabilidad y los controles permiten preparar a los agentes para generar código de valor, incorporando skills, datos del proyecto, acceso a Git y acceso a los pipelines de CI/CD para validar calidad y seguridad.



Expectativas - Trazabilidad total

Toda decisión queda registrada.

QUÉ REGISTRAMOS

- Prompts enviados (versionados)
- Modelo y parámetros usados
- Resultados del Agente Juez
- Ejecuciones de CI/CD asociadas
- Artefactos generados

► *Para responder a cualquier auditoría con evidencias, no con suposiciones.*



Recuperación – Validación real

Reproducible bit a bit.

QUÉ GUARDAMOS

- Historial completo de la sesión
- Snapshot de contexto al ejecutar
- Salida y veredicto del Juez
- Estado del pipeline en ese commit
- Versión exacta del modelo

► *Ante incidente o regresión: rehacemos la ejecución exactamente, sin adivinar.*



Seguridad - Controles incluidos

Calidad y coste, no solo funcionalidad.

QUÉ EVALUAMOS

- Tests automáticos + cobertura
- Análisis estático (SonarQube)
- Dependency checks · CVEs
- Coste por modelo y por ejecución
- Decisión final del Juez trazada

► *Resultados automatizados y rastreables hasta el origen.*

CASO DE USO

Realizar una API en FASTAPI

QUÉ VAS A VER

1

La UI de ALTEN

Interfaz donde se gestionan tareas y revisiones.



2

Se asigna una tarea

Descripción de lo que el agente debe generar.



3

Aparecen los HITL

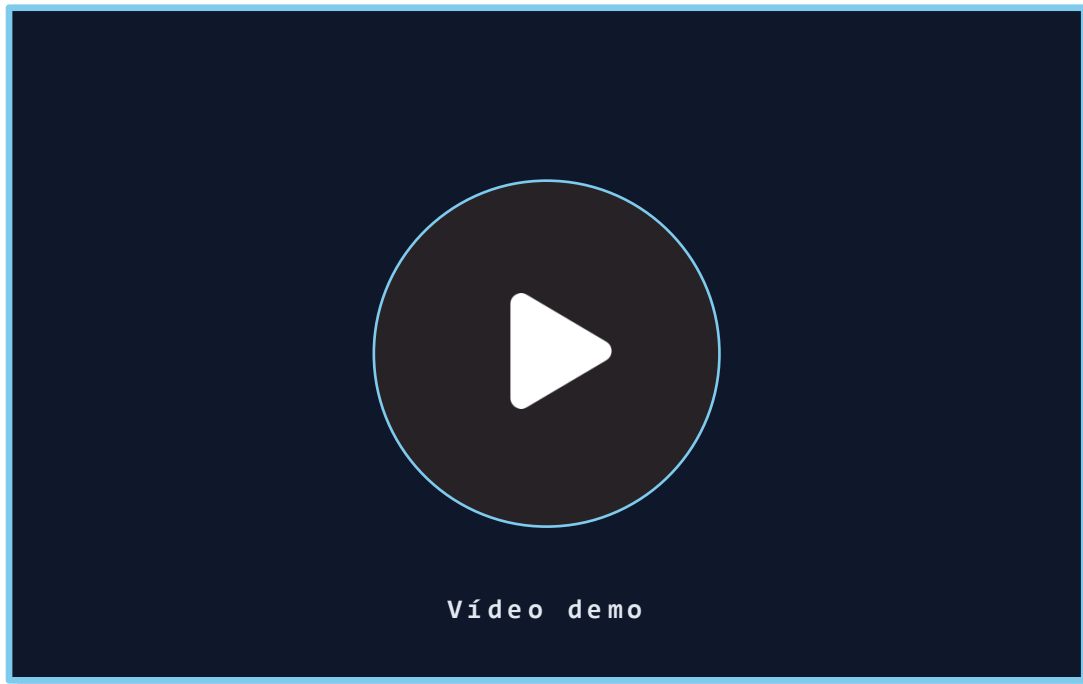
Puntos de revisión humana antes de continuar.



4

Código validado

Resultado final que ha pasado todos los controles.



PRÓXIMOS PASOS

Hacia un paradigma agéntico del SDLC

La implantación de agentes en el desarrollo de software no sigue un único camino, pero sí permite identificar próximos pasos recomendados para avanzar con cierto criterio:



EMPIEZA PEQUEÑO

Pilota antes de escalar.

Selecciona proyectos, plataformas y frameworks.
Experimenta, mide costes, evita sorpresas.



CONTEXTO ANTE TODO

Sin contexto, los agentes son inútiles.

Alimenta al sistema con tu spec, arquitectura, convenciones y vocabulario de dominio.



HUMAN-IN-THE-LOOP

No es opcional. Es estructural.

Define checkpoints de revisión humana antes de que cualquier output llegue a producción.



MIDE, CONSTRUYE, AJUSTA

Si no lo mides, no lo mejoras.

Cobertura real, falsos positivos, tiempo ahorrado, defectos escapados. Datos antes que opiniones.



expoQA[®]26

MADRID 26th, 27th & 28th May

Thank you for attending

expoqa.eu