

# QUANTUM INSPECTOR

Sistema integrado de inspección QA para HMI en Android Automotive

**Autor:** David Erik García Arenas

**Organización:** Paradox Cat GmbH

**Contexto:** Android Automotive · HMI · Herramientas de QA

**Tipo de documento:** Whitepaper técnico interno

**Estado:** Prueba de concepto funcional (PoC)

**Confidencialidad:** Uso interno

## Resumen ejecutivo

El aseguramiento de la calidad en interfaces Hombre-Máquina (HMI) sobre Android Automotive implica trabajar simultáneamente con múltiples artefactos distintos: lo que se ve en pantalla, la jerarquía real de la interfaz, sus atributos estructurales, los selectores utilizados en automatización y las señales de ejecución del sistema.

En la práctica actual, estos artefactos se gestionan mediante herramientas independientes, lo que da lugar a flujos de trabajo fragmentados, una elevada carga cognitiva y una fuerte dependencia del razonamiento manual del ingeniero para reconstruir el contexto correcto en cada iteración.

Este documento presenta QUANTUM Inspector, una prueba de concepto funcional desarrollada para validar una hipótesis concreta: que en QA de HMI la eficiencia y la calidad no están limitadas principalmente por la potencia de cada herramienta individual, sino por la falta de integración contextual entre ellas.

QUANTUM propone unificar la inspección visual, el análisis estructural de la interfaz y el razonamiento QA en una única interfaz sincronizada. Los resultados observados durante el PoC indican mejoras claras en fluidez operativa, reducción del cambio de contexto y mayor robustez en la construcción de selectores.

Estos resultados deben interpretarse como tendencias observadas en un entorno controlado de validación, no como métricas definitivas de un sistema productivo.

## 0. Alcance del PoC.

QUANTUM Inspector se presenta explícitamente como una prueba de concepto orientada a validación, no como una herramienta productiva cerrada.

El PoC demuestra de forma funcional:

- correlación visual y estructural entre píxel y nodo de interfaz.
- inspección sincronizada de captura visual y jerarquía XML.
- análisis y revisión de snapshots en modo offline.
- heurísticas de apoyo a la generación y evaluación de selectores.

Quedan fuera del alcance actual del PoC:

- integración directa con pipelines productivos de integración continua.
- escalado multiusuario o multirrol.
- validación exhaustiva en todas las configuraciones OEM.
- automatización completa de decisiones de QA.

Este encuadre es intencional y garantiza que los resultados presentados se interpreten en el contexto adecuado.

## Disponibilidad del PoC.

El código fuente correspondiente a este PoC se encuentra publicado en un repositorio GitHub con fines de revisión técnica y discusión interna:

<https://github.com/Hawaiiiiiii/qa-snapshot-inspector-poc>

El repositorio refleja el estado actual del PoC y su propósito es facilitar la evaluación del enfoque y del diseño técnico. No está concebido como herramienta productiva ni como referencia final de arquitectura. El repositorio no incluye dependencias específicas de entorno OEM ni configuraciones internas, y se limita a ilustrar el diseño conceptual y funcional del sistema.

## 1. Introducción.

Las HMI modernas basadas en Android Automotive constituyen sistemas altamente dinámicos caracterizados por:

- jerarquías de vistas profundas y cambiantes.
- gestión de foco y eventos mediada por overlays complejos.
- pipelines de renderizado en tiempo real.

- cambios frecuentes de estado.
- requisitos de calidad estrictos impuestos por OEMs.

En el día a día, los ingenieros de QA se ven obligados a responder de forma recurrente a una serie de preguntas muy concretas:

- qué elemento se está observando realmente en pantalla.
- a qué nodo de la jerarquía corresponde.
- cómo puede referenciarse de forma estable.
- qué información de ejecución explica su comportamiento.

Aunque la complejidad del sistema ha aumentado de forma notable, las herramientas empleadas para responder a estas preguntas siguen operando de manera aislada, obligando al ingeniero a reconstruir manualmente el contexto en cada iteración.

QUANTUM Inspector surge como respuesta a esta ineficiencia estructural, no para sustituir el criterio del ingeniero, sino para ayudar a preservarlo a lo largo del proceso de inspección.

## 2. Planteamiento del problema: inspección QA fragmentada.

### 2.1 Flujo de trabajo tradicional.

Un ciclo típico de inspección QA en Android Automotive incluye:

1. captura visual desde el dispositivo o rack.
2. exportación del volcado XML de UIAutomator.
3. apertura del XML en un visor independiente.
4. localización manual del nodo correspondiente.
5. copia de atributos relevantes.
6. construcción manual de selectores.
7. consulta paralela de logs.
8. iteración por prueba y error.

Este flujo **introduce tres problemas** estructurales claros:

- fragmentación de herramientas.
- elevado cambio de contexto.

- razonamiento manual propenso a errores.

## 2.2 Coste cognitivo y operativo.

Desde la perspectiva de la interacción humano-computadora y la psicología cognitiva aplicada al trabajo técnico:

- el cambio frecuente de tarea introduce penalizaciones medibles de tiempo y error.
- el aumento del número de alternativas incrementa el tiempo decisional (*ley de Hick-Hyman*).
- los flujos fragmentados generan fatiga y reducen la eficacia ejecutiva.

Estos costes rara vez aparecen reflejados en métricas brutas, pero se acumulan de forma significativa en entornos QA reales.

## 3. QUANTUM Inspector: visión general del sistema.

### 3.1 Principios de diseño.

QUANTUM Inspector se fundamenta en tres principios:

1. **contexto único:** un único espacio mental y visual sincronizado.
2. **correlación inmediata:** vínculo directo entre píxel, nodo y estructura.
3. **reducción de decisiones periféricas:** apoyo heurístico al ingeniero sin sustituir su criterio.

### 3.2 Capacidades validadas en el PoC.

El PoC integra de forma funcional:

- mirroring en vivo del dispositivo.
- visualización jerárquica de la interfaz.
- mapeo píxel a nodo.
- generación asistida de selectores.
- heurísticas de evaluación de robustez.
- ascenso al nodo clicable relevante.
- inspección contextual de logs.
- revisión offline mediante snapshots.

### 3.3 Modelo de evaluación de selectores.

Los selectores se clasifican de forma heurística en tres categorías:

- **unique**: estabilidad alta esperada.
- **ambiguous**: posibles colisiones.
- **fragile**: dependencia de índices o layout.

Este modelo no automatiza decisiones, pero hace explícita la calidad del selector, reduciendo errores aguas abajo en automatización.

## 4. Metodología de evaluación.

### 4.1 Diseño.

La evaluación se realizó como comparación A/B en un entorno controlado de QA HMI durante un periodo de validación del PoC.

La unidad de análisis fue la iteración completa de inspección, definida como la obtención de:

- confirmación visual.
- identificación correcta del nodo.
- selector utilizable.
- contexto de depuración suficiente.

### 4.2 Métricas consideradas.

Se observaron tres familias de métricas:

- métricas operativas (tiempo de ciclo, fluidez).
- métricas de calidad (robustez del selector, errores detectados).
- métricas cognitivas (cambio de contexto, carga decisional percibida).

Estas métricas se interpretan como indicadores de tendencia, no como valores absolutos extrapolables.

## 5. Resultados observados.

### 5.1 Eficiencia operativa.

Durante el PoC se observó:

- reducción consistente del tiempo necesario para completar una iteración.
- aumento del número de inspecciones posibles por unidad de tiempo.

- menor necesidad de repetir iteraciones por errores de correlación.

Los valores concretos varían según escenario, pero la tendencia es estable.

## **5.2 Calidad del resultado.**

Se observó:

- menor generación de selectores frágiles.
- mayor visibilidad de riesgos antes de integrar automatización.
- reducción de errores derivados de correlación manual.

## **5.3 Eficiencia cognitiva.**

La unificación del flujo produjo:

- menor cambio de herramientas por iteración.
- reducción de decisiones periféricas.
- mayor continuidad del razonamiento técnico.

# **6. Discusión.**

## **6.1 Valor de la integración.**

El principal valor de QUANTUM no está únicamente en la velocidad, sino en la preservación del contexto mental durante la inspección.

Al colapsar artefactos dispersos en un único espacio coherente, el ingeniero puede concentrarse en el problema real, no en la mecánica de la inspección.

## **6.2 Implicaciones para equipos de QA.**

Como efecto colateral positivo, se observó:

- reducción de la curva de onboarding.
- externalización del conocimiento experto en artefactos inspectables.
- menor dependencia de memoria individual o experiencia implícita.

# **7. Limitaciones y trabajo futuro.**

**Limitaciones actuales:**

- alcance restringido al contexto de PoC.
- heurísticas simples, no modelos predictivos.

- gestión manual de baselines.

#### **Líneas futuras potenciales:**

- integración progresiva con sistemas de integración continua.
- análisis de diffs visuales y estructurales asistidos.
- modos colaborativos de revisión.
- instrumentación de métricas de uso.

Estas líneas representan posibilidades de evolución, no compromisos de implementación.

## **8. Conclusión.**

QUANTUM Inspector valida una hipótesis clara: que en entornos HMI de alta complejidad la eficiencia y la calidad del QA dependen más de la integración contextual entre herramientas que de la potencia aislada de cada una de ellas.

Como prueba de concepto, QUANTUM demuestra que un enfoque integrado puede:

- reducir el tiempo de inspección.
- mejorar la calidad de los selectores.
- disminuir la carga cognitiva.
- escalar la capacidad QA sin aumentar el tamaño del equipo.

Este documento no presenta una solución cerrada, sino una base sólida y realista sobre la que evolucionar tooling interno de QA.

## **Agradecimientos.**

Este trabajo se ha desarrollado en el contexto de operaciones reales de QA en Android Automotive dentro de Paradox Cat GmbH.

Agradezco a **Leandro Lourenço Miranda** por la orientación técnica, el feedback crítico y el contexto profesional proporcionado durante el desarrollo del PoC.