



Connect

De asistir a ejecutar: el salto del copiloto al ciclo de desarrollo agéntico

Daniel Forcén Esteban

Responsable de Arquitectura e Innovación

Sorteo de dos plazas de formación en IA

Entre los asistentes a la sesión de Grupo Carreras “De asistir a ejecutar: el salto del copiloto al ciclo de desarrollo agéntico” se sortearán dos plazas de formación en IA que incluyen:

- ▶ Asistencia al **curso** Developing and Deploying AI/ML Applications on Red Hat OpenShift AI (AI267), en formato virtual el 9 de noviembre.
- ▶ Un año de **acceso** a la plataforma de eLearning (RHLS) para ese mismo curso.
- ▶ Un **voucher** de examen gratuito.

El sorteo se realizará en fechas posteriores al Connect y Red Hat se pondrá en contacto directamente con los ganadores.





Daniel Forcén Esteban

Responsable Arquitectura e Innovación
Carreras Grupo Logístico



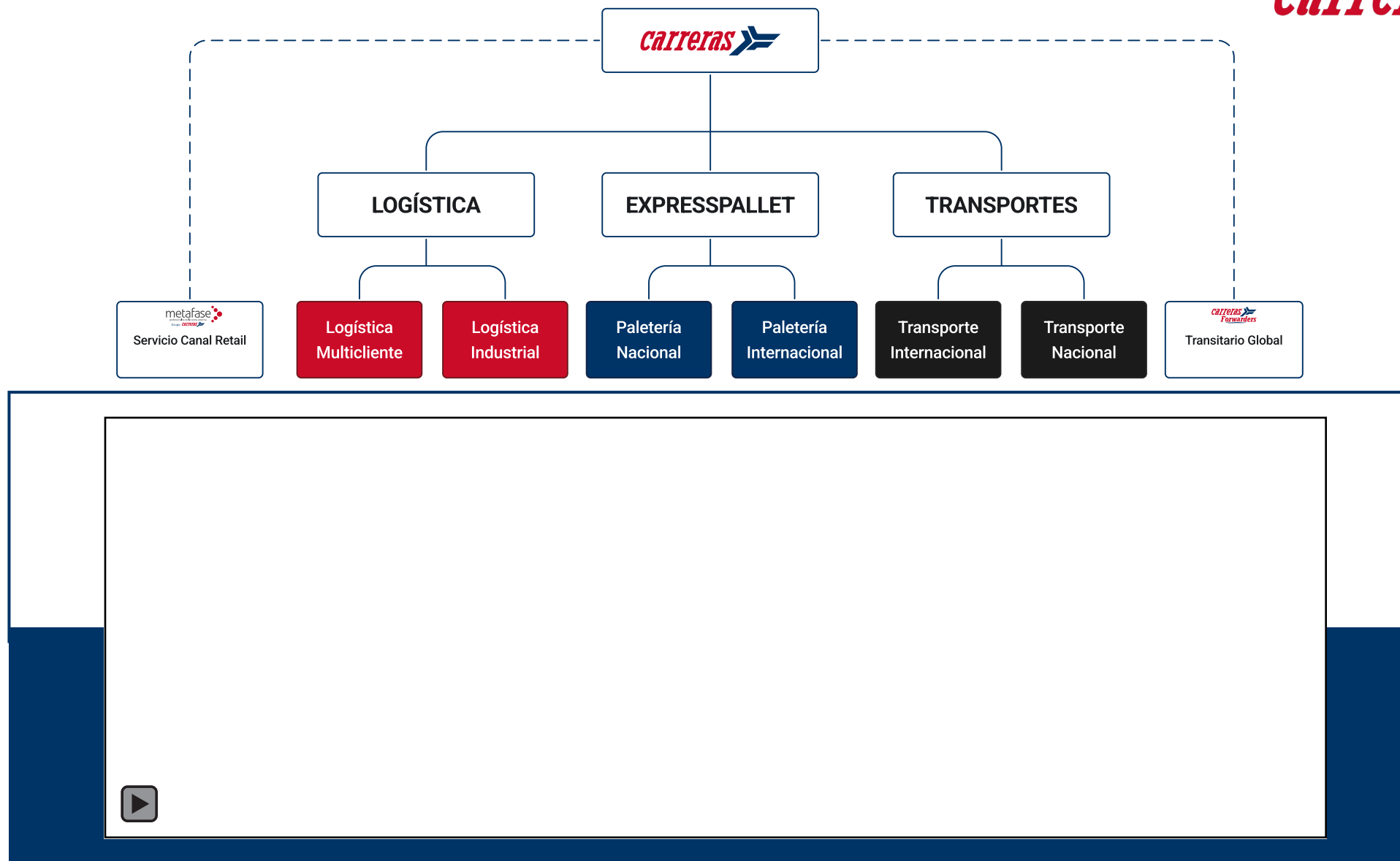
DISCLAIMER

Esto no es marketing Está en producción

Desde el 3 de julio · hecho por y para Carreras

Jul 2026, Aug 2026 ▾		
		Issues resolved
Ada	+ Jul 2026	226
	+ Aug 2026	528







+2.800

Personas trabajando



+140 IT

España + Argentina



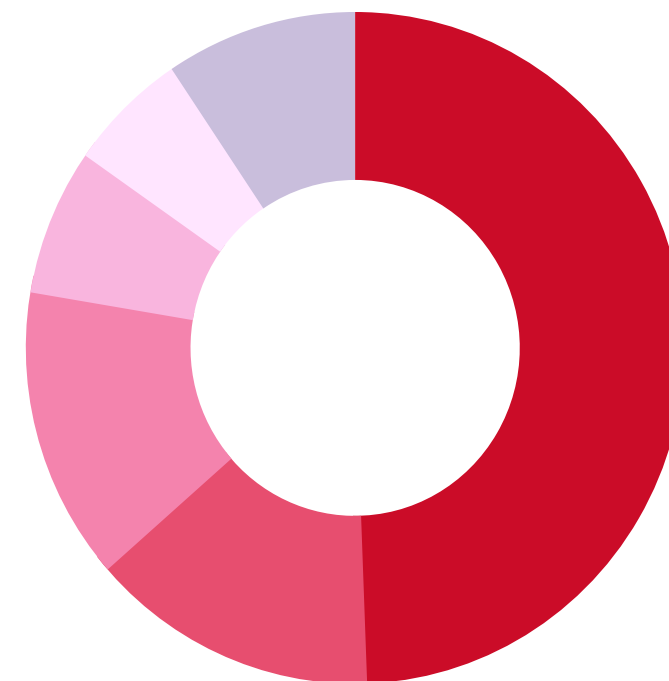
+2.000

Vehículos controlados



+670.000 m²

Instalaciones logísticas

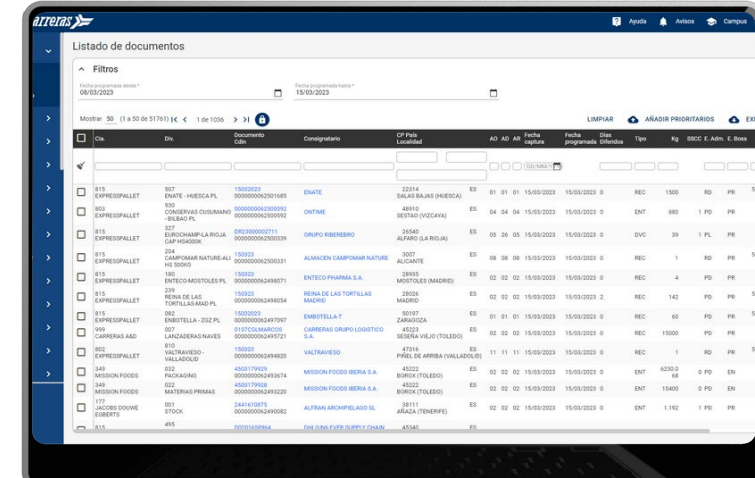
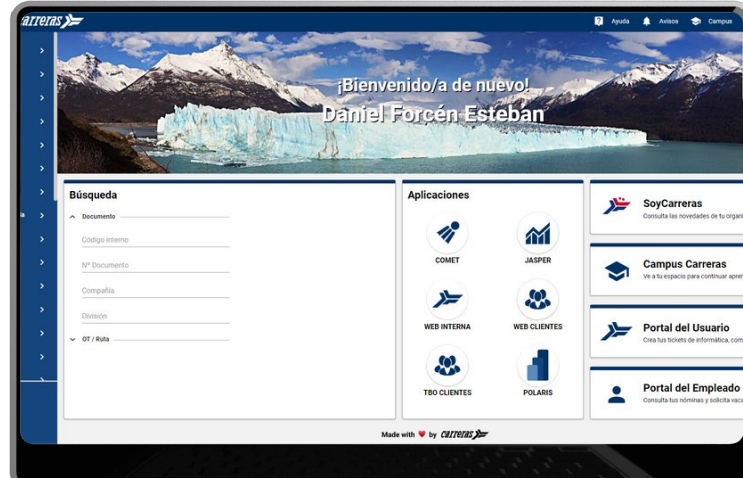


No desarrollo
50,6%

Desarrollo
49,4%



TBO



APPs



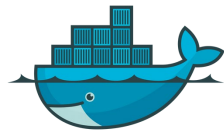
Swift



mongoDB®



OPENSIFT

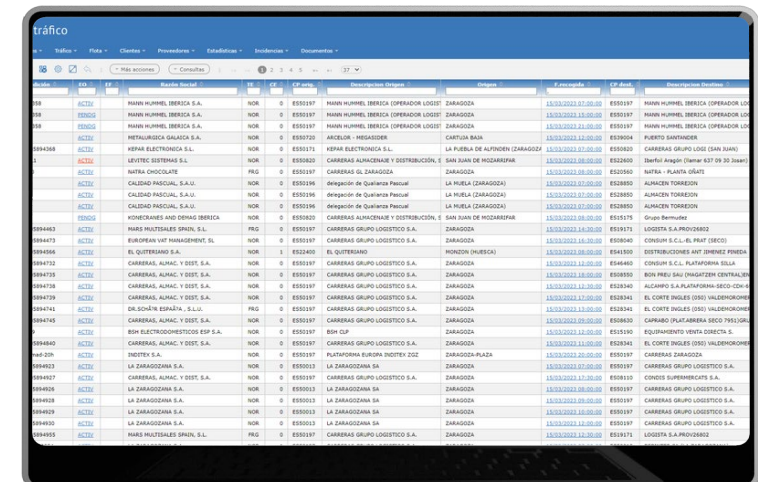


docker

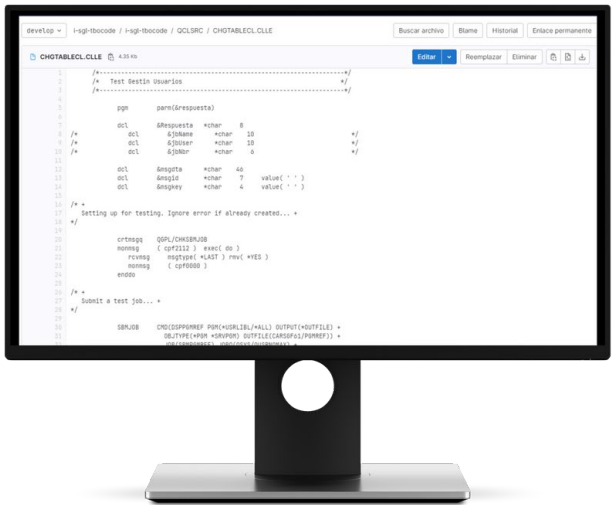
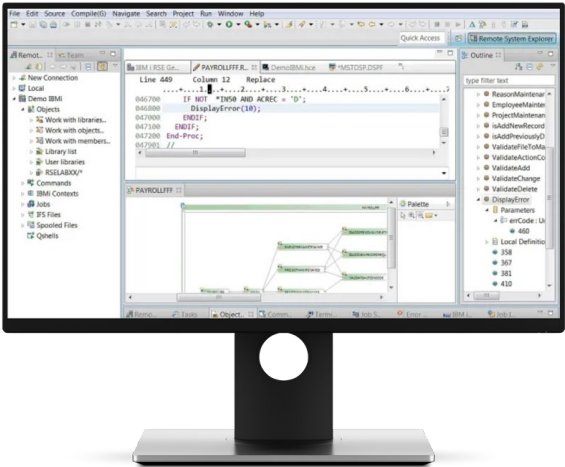


Kotlin

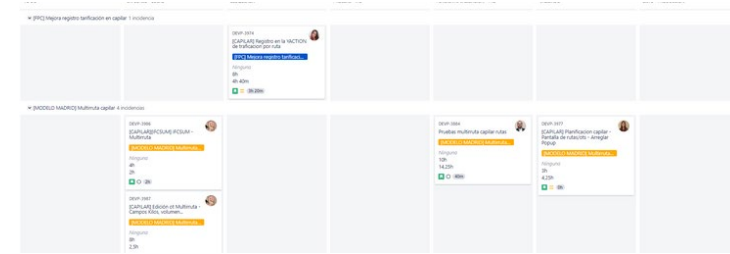




IBM AS/400



Planificación



Integración

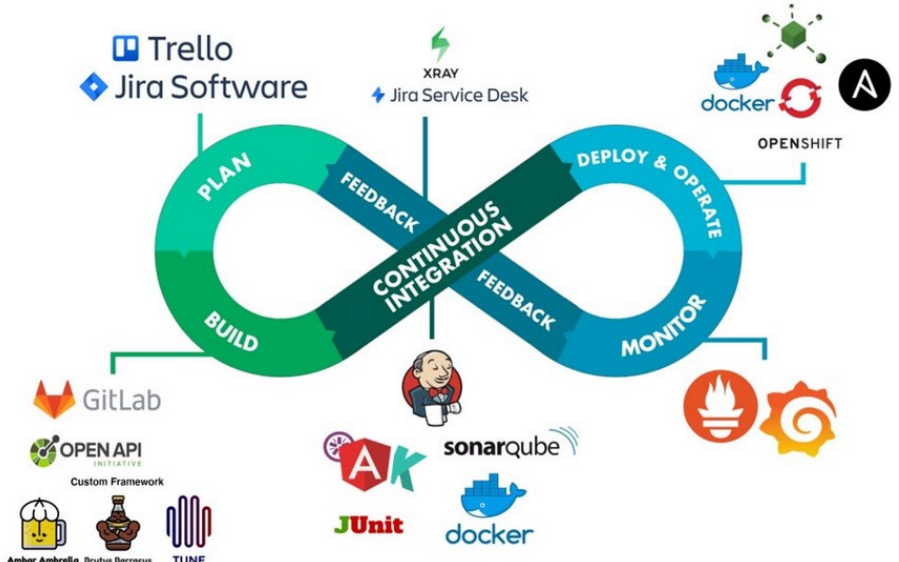
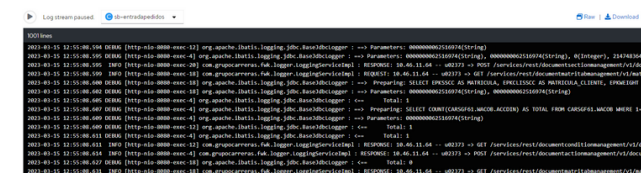
Stage View

	[SETUP]	[MAVEN] Test	[MAVEN] Build	[SONAR] Scanner	[SONAR] Quality Gate check	[DOCKER] Build & Push	[OCP] Update Imagestream
Average stage times:	7s	32s	12s	7s	59ms	13s	1s
(Average full run time: ~ 1min 23s)							
#14.15	4s	29s	12s	8s	55ms	13s	1s
Mar 15 10:02							
#14.14							

Despliegue



Monitorización



Evolución del ciclo de vida de desarrollo

As Is



Modelo estándar con transferencias entre roles en metodología agile

De 7 roles secuenciales a 3 capacidades integradas asistidas por IA

To Be



Skills + Sistema agéntico + Observabilidad/Ciberseguridad + Automatización inteligente



ADA – Factoría Agéntica



El coste de un evolutivo pequeño no está en pensar

- **Un evolutivo repite siempre los mismos pasos**

Análisis, tareas en JIRA, rama, código backend, código frontend, tests, seguridad, MR, documentación y evidencias de compliance.

- **Casi ninguno de esos pasos requiere criterio: requieren no equivocarse**

Campos obligatorios, convenciones de rama y commit, separación de duties, trazabilidad de la épica al MR. Es trabajo exacto, no trabajo creativo.

- **Asegurar la trazabilidad y cumplimiento de los desarrollos**

Todas las tareas están vinculadas con GitLab y un documento funcional

- **Medir el coste de cada tarea en tiempo y dinero**

Podemos medir en tiempo y € cada tarea que ejecuta la IA

Objetivo: automatizar el ciclo completo delegando al modelo solo el razonamiento, y manteniendo en código todo lo que debe ser exacto.



Dos piezas, una responsabilidad cada una

ada-agents

EL QUÉ Y EL CÓMO

Repositorio de configuración. No contiene producto: contiene los agentes, sus prompts, el conocimiento de cada stack y un CLI que ejecuta todo lo que no debe decidir un modelo.

22 agentes · 19 prompts · 35 comandos

ada-runner

EL DÓNDE Y EL CON QUÉ

Imagen de contenedor que ejecuta esos agentes contra los repos corporativos. Prepara el contexto, llama al modelo, mide lo que cuesta y devuelve el resultado a JIRA y GitLab.

5 runtimes · 4 imágenes · 54 tests

Están separados a propósito: el conocimiento cambia a diario, la infraestructura no. Cambiar un prompt no reconstruye una imagen.



Cómo encajan las piezas



ada-runner

El contenedor, el setup corporativo (CA, Nexus, Git), la preparación del contexto, la llamada al modelo y la contabilidad de coste.

ada-agents

Qué agentes existen, qué sabe cada uno, qué fases corre cada tipo de proyecto y qué operaciones se hacen por código y no por prompt.

El contenedor no sabe de negocio y la configuración no sabe de infraestructura. Se encuentran en tiempo de ejecución.



ADA - agents



Un agente por responsabilidad, no uno que lo haga todo

Agente	Responsabilidad	Entrega
orchestrator	Coordina el pipeline. Único que invoca subagentes; nunca escribe código	status.json · informe final
analyst	Análisis funcional, modelo de datos, tareas JIRA y Confluence	analysis.md · tasks-spec.md
backend / frontend	Implementación en Java 21 + Spring Boot y en Angular 21	backend / frontend-report.md
testing-backend	Pruebas unitarias con JUnit 5 y Mockito	testing-backend-report.md
security-*	SAST, SCA y secretos. Bloquea si hay severidad crítica o alta	security-*-report.md
validation-*	Compila, hace push y crea el Merge Request	*-validation-report.md
qa-e2e	Integración con Playwright sobre el repo E2E compartido	testing-e2e-report.md
compliance	Trazabilidad ISO 27001: separación de duties y evidencias	compliance-report.md
diagnosis x4	Incidencias en solo lectura: RPG, Java EE, web y móvil	diagnosis-report.md



El pipeline de un evolutivo web



web-application · 10 fases

El ciclo completo, con frontend y pruebas E2E. 80 de los 96 proyectos.

java-ee-application · 6 fases

Sin frontend ni E2E. El monorepo Comet.

*-diagnosis · 1 fase

Solo investigación en lectura. Sin rama, sin worktree, sin MR.

Las fases no se deciden en el prompt: las declara project-types.json según el tipo de proyecto.



Los agentes no se hablan entre ellos

Cada fase lee los informes de las anteriores y escribe el suyo. El único canal es el sistema de ficheros, en `.pipeline/{ÉPICA}/`.

Eso tiene tres consecuencias prácticas:

- Cada fase es reproducible y auditable por separado.
- Un fallo se acota a una fase, no contamina el resto.
- El estado vive en `status.json`, no en la memoria del modelo.

.pipeline/DID-16449/

<code>status.json</code>	estado y fases
<code>epic-info.md</code>	contexto funcional
<code>analysis.md</code>	análisis del analista
<code>tasks-spec.md</code>	tareas de alto nivel
<code>backend-report.md</code>	qué implementó
<code>frontend-report.md</code>	qué implementó
<code>security-*.md</code>	hallazgos
<code>compliance-report.md</code>	evidencias ISO
<code>usage.json</code>	coste por fase
<code>incidents/</code>	fallos irre recuperables

Tres intentos por track, backend y frontend independientes. Agotados, se abre un incidente en lugar de entregar algo dudoso.



Lo que es determinista no se le pide al modelo

3.706 líneas de Node que sustituyen a los MCP de JIRA y Confluence en toda operación de escritura.

Pipeline • 9

init, status, codemap, preflight, diff-summary, copy-docs, cleanup

JIRA • 16

epic, create-epic, init-initiative, create-tasks, complete-task, cost

GitLab y Git • 5

push con rebase, verify-diff, create-mr, link-mr, search

Confluence • 3

publish, link, search

Worktrees • 2

create, remove

Fijado por código

Proyecto • tipo Historia • rama feature/{KEY}
• MR a env/dev • campos de IA y de coste

El modelo ya no puede olvidar un campo obligatorio ni inventar una convención: no es él quien los escribe.



ADA - runner



Una imagen, cinco motores de IA

Runtime	Credencial	Configuración
bob	BOBSHELL_API_KEY	bob-full → .bob
kiro	KIRO_API_KEY	kiro-full → .kiro
claude	ANTHROPIC_API_KEY	claude-full → .claude
codex	OPENAI_API_KEY	codex-full → .codex
opencode	OPENCODE_API_KEY	opencode-full → .opencode

▼ Detalles clave

Field Tab [IA](#)

AI Assistant Used

Kiro

AI Contribution Perc100 %

Coste IA0,27 €

```
agent_run <fase> <prompt> <captura> [modo]
→ bob_run · kiro_run · claude_run · codex_run · opencode_run
```



El cuello de botella se mueve

Cuando construir deja de ser el paso lento, el límite pasa a estar en validar, priorizar y en la capacidad de negocio para responder.

Eso reordena el modelo operativo:

- Los arquitectos pasan de escribir código a afinar agentes y su conocimiento.
- Los desarrolladores pasan de implementar a validar y desplegar.
- El activo estratégico deja de ser el código y pasa a ser el sistema de agentes.

OBJETIVOS DEL MODELO

≤ 20 días

Time to market de un evolutivo, percentil 80

≥ 70 %

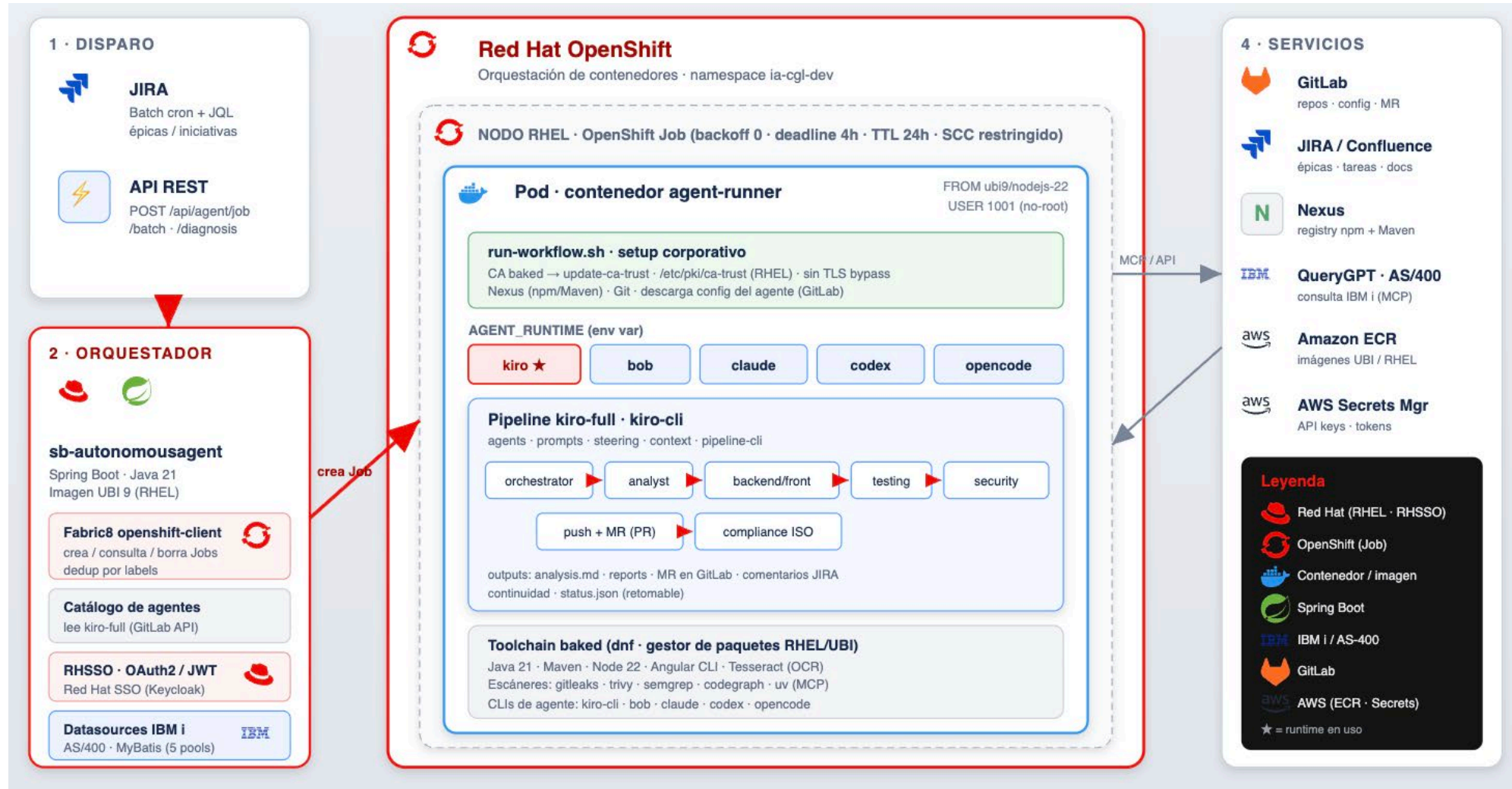
Ejecuciones sin intervención humana

≥ 80 %

Merge Requests aprobados a la primera



Ciclo completo



Impacto real

Todo proceso que tiene un coste necesita ser medido y calculado su rendimiento

- 130+ Merge Request publicados
- 20 horas de un desarrollador vs 0,25\$ por evolutivo y/o correctivo
- Aproximadamente tarda 1 hora en realizar el evolutivo
- Actualmente gestionado con la licencia de 200€/mes de Kiró
- 40 días SLA vs 9 días SLA
- Cuando el contexto no es claro no se puede cumplir el SLA

Evolutivo pequeño

SLA: 20 días



■ Dentro de SLA ■ Fuera de SLA

233

Total

9

Media días

212

Dentro SLA

21

Fuera SLA



¿Y... el contexto?



QueryGPT

Lenguaje natural → SQL · AS/400 DB2 for i



Los datos están ahí. Falta el contexto

- **Tablas con nombres poco representativos**

El AS/400 limita los nombres a diez posiciones. CABEXP, SGD006, FACPRD. No son fáciles de identificar por el nombre

- **Claves ajenas no obligatorias**

Las relaciones existen en la cabeza de las personas y en el código RPG, no en el catálogo.

- **Tipos que pueden no ser lo que parecen**

Las fechas se guardan como DECIMAL(8,0) en formato AAAAMMDD. Un JOIN ingenuo o un BETWEEN mal escrito devuelven un resultado plausible y falso.

- **La lógica de negocio vive en los programas**

Miles de fuentes RPG y CLs programadas. La regla que explica un campo calculado está enterrada en una subrutina.

Cada pregunta de negocio necesita un análisis para identificar de donde obtener el dato y su lógica. Ese es el cuello de botella, no la base de datos.



EL PROBLEMA QUE RESOLVEMOS

De la pregunta al dato

1 Preguntas

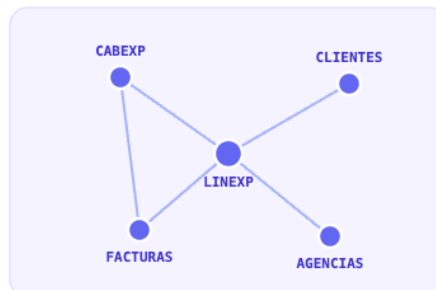
Escribes en tu idioma, como en un chat. Sin saber SQL.

«¿Cuántas expediciones facturó Barcelona durante agosto?»

LENGUAJE NATURAL

2 Entiende el negocio

Grafo de conocimiento: tablas, JOINs, reglas y vocabulario.



3 Genera el SQL

Consulta correcta y auditable, con el plan de ejecución.

```

SELECT  c.DELEGACION,
        COUNT(*) AS expediciones,
        SUM(f.IMPORTE) AS total
FROM    DATOEXP.CABEXP c
JOIN    DATOEXP.FACTURAS f
        ON f.NUMEXP = c.NUMEXP
WHERE   c.FECHA BETWEEN '2026-08-01'
        AND '2026-08-31'
GROUP BY c.DELEGACION
  
```

4 Devuelve el dato

Ejecuta contra el AS/400 y muestra el resultado real.

DELEGACIÓN	EXPED.	TOTAL €
BARCELONA	1.284	412.905
MADRID	1.107	388.240
VALENCIA	642	201.815

● AS/400 - DB2 for i



Servidor MCP

17 herramientas para IDEs y agentes IA



Análisis de impacto

Qué se rompe si cambias una tabla



Memoria de código

Búsqueda semántica en los repositorios



Ejemplos validados

Aprende del SQL que ya funciona



De tres mil tablas a sesenta, sin preguntar al modelo

Meter tres mil esquemas en un prompt es imposible y caro.

1

Señales de texto sobre el catálogo sin coste

Búsqueda de texto completo sobre nombres, alias de negocio y descripciones, en pasada estricta y pasada laxa, con puntuación acumulativa y refuerzo para el workspace del dominio.

2

Similitud con consultas ya resueltas sin coste

Las preguntas parecidas ya respondidas aportan sus tablas como candidatas. Los ficheros lógicos se descartan: solo entran tablas físicas, vistas y alias.

3

El modelo elige entre las finalistas con coste

Sobre sesenta u ochenta candidatas, un agente selecciona las tablas definitivas; si la pregunta es ambigua, pide aclaración en lugar de adivinar. Después, tres tareas en paralelo: poda de columnas, ejemplos validados y contraejemplos de consultas que fallaron.



Un grafo de todo lo que se sabe del dominio

Se construye desde	Produce	Detalle
Catálogo de tablas y columnas	Nodos de tabla, vista, fichero lógico y columna	El tipo de nodo se deriva del tipo de objeto del AS/400
Relaciones inferidas	Aristas de clave ajena	Con peso según la confianza de la inferencia y la condición de JOIN guardada
Programas fuente analizados	Nodos de programa	Solo los relevantes; se descartan los de menos de veinte líneas
Mapa de lectura y escritura	Aristas de escribe y de lee	Con peso distinto según sea escritura, actualización, borrado o lectura
Reglas de negocio extraídas	Nodos de regla y aristas de afecta a	Enlazadas a las tablas concretas que gobiernan
Índices existentes	Nodos de índice	Con sus claves y su número de usos reales

El código se indexa y se guardan las relaciones reduciendo tiempo y coste



Quién llama a quién, sin mirar un diagrama

Diecisiete patrones que detectan el acoplamiento real entre repositorios leyendo el código, no la documentación.

Llamadas HTTP

Cientes declarativos, plantillas REST, clientes reactivos y los endpoints publicados en el otro extremo.

Mensajería

Envíos y escuchas de Kafka y de RabbitMQ, incluidos los nombres de tema definidos como constantes.

Base de datos compartida

Entidades que apuntan a la misma tabla desde servicios distintos, que es el acoplamiento más difícil de ver.

Llamadas al AS/400

Programas invocados desde RPG, llamadas a procedimientos y URLs de servicios consumidas desde el legacy.

El truco está en emparejar los dos extremos: se construye un índice de productores y consumidores por nombre de tema, así que publicar y consumir dejan de ser dos hechos aislados y pasan a ser una arista.



Todo esto es una herramienta para otros agentes

Familia	Herramientas	Qué permite
Consulta SQL	3	Ejecutar, explicar el plan y listar sistemas
Modelo de datos	8	Workspaces, tablas, grafo, camino de JOIN, exploración
Generación	3	Generar SQL, generar y ejecutar, historial
Memoria de código	6	Buscar código, recordar, indexar, relaciones, enrutar
Localizador funcional	2	Localizar e indexar funcionalidades
Criticidad y dependencias	5	Calcular riesgo, listar críticas, dependencias implícitas
Linaje de columna	3	Extraer, trazar hacia atrás y hacia delante
Análisis e histórico	7	Impacto, análisis de fuentes, cambios recientes, documentación

Es el sistema nervioso de la fábrica de software. El agente funcional consulta el modelo y los de implementación precedentes



Aprendizajes

- **OOMKilled por el heap de Node contra la RAM del nodo**

Al compilar en paralelo el job se quedaba sin memoria.

- **Evitar duplicar jobs**

Detectar por nombre o tag que una ejecución está en curso y cambiar de estados para que no lance la misma iniciativa N veces.

- **Lo determinista fuera del LLM**

Para asegurar el mismo funcionamiento y ahorrar tokens crear scripts y no delegar en el modelo.

- **El orquestador se paraba porque requería confirmación**

Reforzar el prompt para que sea capaz de tomar decisiones, aunque sean incorrectas, pero que no bloquee el job

- **Medir es lo más importante**

Sin medidas no somos capaces de mejorar el ciclo

Es importante medir y estar en contacto con el resultado y desarrolladores para que sea un trabajo retroactivo



Y Red Hat... ¿qué nos aporta?

Todo desarrollo necesita una buena infraestructura que lo soporte

1

OpenShift para job efímeros

Posibilidad de cómputo escalable y efímero para el lanzamiento de procesos paralelizables

2

DeveloperHub

Inventario de aplicaciones y relación entre ellas para aumentar el contexto

3

OpenShift AI... próximamente

Cambio de modelo de pago por uso a pago por la infraestructura





DEMO REDHAT SUMMIT



✓ Detalles clave

Field Tab IA

Tipo de iniciativa

Evolutivo

Informador

 Daniel Forcen Esteban

Solicitante

Daniel Forcen

Origen de la petición

DIT

Empresa

CGL

Sponsor

Añadir texto

Área responsable

Arquitectura

Aplicación afectada

 Itapp



Descripción

Quiero que generes una pantalla donde se vea un camión de Carreras, el logo de RedHat y abajo ponga Carreras loves Redhat con un corazón y abajo Gracias Redhat Summit 2026



Lanzar Desarrollo Automático

Introduce la clave de la épica, iniciativa o tarea (IN/DID) y selecciona el agente.



Issue Key (ej: INI-123, DID-456)

IN-7925

Agente

Orchestrator

Pipeline orchestrator. This is the default AGENT_NAME used by agent-runner. Receives a JIRA epic and coordinates the full pipeline execution invoking specialized subagents. NEVER implements code directly.

Sistema

TBO

Motor

KIRO — pipeline web (Java/Spring + Angular)

El motor es independiente del sistema. Al cambiarlo se recarga la lista de agentes, porque cada uno declara los suyos.

Nombre del Proyecto (opcional para iniciativas IN-*)

itapp



Pods > Pod details

P kiro-in-7925-orchestrator-itapp-djrrz Running

★ Actions ▾

Details Metrics YAML Environment **Logs** Events Terminal



agent-runner ▾

Options ▾

Search logs

Current log ▾



296 lines

```

213 2026-09-11T09:37:03.398Z
214 2026-09-11T09:37:04.613Z 👤 DID-16906 asignado a Ada
215 2026-09-11T09:37:04.925Z 🔄 Transicionando DID-16906 → En progreso...
216 2026-09-11T09:37:05.465Z ✅ DID-16906 → En progreso
217 2026-09-11T09:37:05.472Z - Completed in 2.74s
218 2026-09-11T09:37:05.472Z
219 2026-09-11T09:37:09.027Z > Now let's check hasFrontend/backendRepo are both set (full-stack project), then create the JIRA phase tasks. This project has qa-integration phase, not testing-e2e - noted f
220 2026-09-11T09:37:14.316Z I'll create the following file: /workspace/.pipeline/DID-16906/pipeline-tasks.json (using tool: write)
221 2026-09-11T09:37:14.316Z
222 2026-09-11T09:37:14.316Z + 1: [
223 2026-09-11T09:37:14.316Z + 2: {"summary": "[Análisis] Análisis funcional", "description": "Análisis funcional de la épica DID-16906", "labels": ["pipeline-phase", "analysis"], "estimate": "1h"},
224 2026-09-11T09:37:14.316Z + 3: {"summary": "[Backend] Implementación", "description": "Implementación backend de la épica DID-16906", "labels": ["pipeline-phase", "backend"], "estimate": "3h"},
225 2026-09-11T09:37:14.316Z + 4: {"summary": "[Frontend] Implementación", "description": "Implementación frontend de la épica DID-16906", "labels": ["pipeline-phase", "frontend"], "estimate": "3h"},
226 2026-09-11T09:37:14.316Z + 5: {"summary": "[Testing] Tests unitarios backend", "description": "Tests JUnit 5 para código backend de DID-16906", "labels": ["pipeline-phase", "testing"], "estimate": "1h"},
227 2026-09-11T09:37:14.316Z + 6: {"summary": "[Seguridad] Revisión backend", "description": "SAST/SCA backend de DID-16906", "labels": ["pipeline-phase", "security"], "estimate": "1h"},
228 2026-09-11T09:37:14.316Z + 7: {"summary": "[Seguridad] Revisión frontend", "description": "SAST/SCA frontend de DID-16906", "labels": ["pipeline-phase", "security"], "estimate": "1h"},
229 2026-09-11T09:37:14.316Z + 8: {"summary": "[Validación] MR Backend", "description": "Push + Merge Request backend de DID-16906", "labels": ["pipeline-phase", "validation"], "estimate": "1h"},
230 2026-09-11T09:37:14.316Z + 9: {"summary": "[Validación] MR Frontend", "description": "Push + Merge Request frontend de DID-16906", "labels": ["pipeline-phase", "validation"], "estimate": "1h"},
231 2026-09-11T09:37:14.316Z + 10: {"summary": "[QA E2E] Tests Playwright", "description": "Generación y ejecución de tests E2E con Playwright para DID-16906", "labels": ["pipeline-phase", "e2e"], "estima
232 2026-09-11T09:37:14.316Z + 11: {"summary": "[Compliance] Auditoria ISO 27001", "description": "Verificación compliance ISO 27001 de DID-16906", "labels": ["pipeline-phase", "compliance"], "estimate":
233 2026-09-11T09:37:14.316Z + 12: ]
234 2026-09-11T09:37:14.316Z
235 2026-09-11T09:37:14.316Z Creating: /workspace/.pipeline/DID-16906/pipeline-tasks.json
236 2026-09-11T09:37:14.316Z - Completed in 0.0s

```



Links

Repositorio GitOps

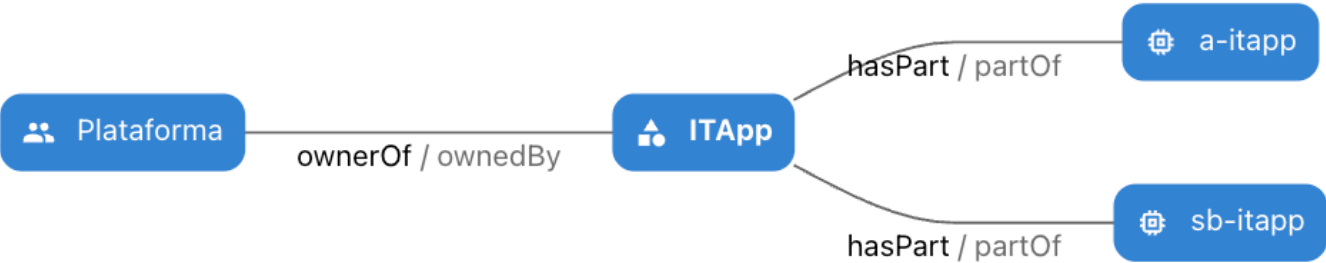
About

VIEW SOURCE VIEW TECHDOCS

DESCRIPTION
Componentes desplegados desde el repositorio de GitOps 'itapp'.

OWNER DOMAIN TAGS
Plataforma No Domain No Tags

Relations



View graph →

Has components

Name	Owner	Type	Lifecycle	Description
a-itapp	Plataforma	website	experimental	Frontend Angular a-itapp....
sb-itapp	Plataforma	service	experimental	Servicio backend sb-itapp...



Data Catalog

Catálogo de tablas, columnas y metadatos inferidos desde DB2 for i

Proyecto: Todos los proyectos Actualizar Sincronizar catálogo

ITAPP

ITAPPDEF
 Definiciones y Categorías de la Aplicación
 4 cols ~56 rows Gestión de Equipos, Puestos y Reservas de Aplicación Interna (ITAPP)

TBODATA

ITAPPEQUIP
 Equipos de Trabajo
 4 cols ~19 rows Gestión de Equipos, Puestos y Reservas de Aplicación Interna (ITAPP)

TBODATA

ITAPPPPL
 Personas / Usuarios de Aplicación Interna
 21 cols ~188 rows Gestión de Equipos, Puestos y Reservas de Aplicación Interna (ITAPP)

TBODATA

ITAPPPSTO
 Puestos de Trabajo de Aplicación Interna
 5 cols ~81 rows Gestión de Equipos, Puestos y Reservas de Aplicación Interna (ITAPP)

TBODATA

ITAPPRESV
 Reservas de Puestos
 4 cols ~261 rows Gestión de Equipos, Puestos y Reservas de Aplicación Interna (ITAPP)

TBODATA

Todas las librerías 5 tablas

TBODATA.ITAPPRESV
 Reservas de Puestos

Documentar con IA Re-sync Cerrar

Documentación IA

Recién generado 9258ms
Regenerar
Exportar .md
X

Tabla: TBODATA.ITAPPRESV — Reservas de Puestos

1. Propósito de la tabla

Esta tabla registra las reservas de puestos (físicos) realizadas por usuarios, identificados mediante su correo electrónico, indicando la fecha en la que se realizó cada reserva. Es utilizada para gestionar la asignación temporal o programada de puestos dentro de la organización.

2. Descripción de columnas

Columna	Descripción de negocio
RESVID	Identificador único de cada registro de reserva. Actúa como clave interna para distinguir cada operación de reserva realizada.
RESVFECH	Fecha correspondiente a la reserva del puesto.
RESVCORREO	Correo electrónico del usuario que realizó la reserva, utilizado como identificador del solicitante.
RESVPSTO	Identificador del puesto reservado. Referencia al puesto físico que fue asignado en la reserva.

3. Relaciones confirmadas

No se proporcionó información de JOINS ni tabla de relaciones explícitas en el contexto. **No se puede determinar sin más contexto** si RESVPSTO se relaciona formalmente con otra tabla de puestos (aunque el nombre sugiere una posible relación con un catálogo de puestos, esto no está confirmado por evidencia en el código o relaciones proporcionadas).

41





Detalles clave

Field Tab IA

AI Assistant Used	Kiro
AI Contribution Porcentaje	100 %
Coste IA	0,21 €

Actividades secundarias

100 % completado

Actividad	Prioridad	Story Poi...	Persona asignada	Estado
TBOARQ-2984 [Análisis] Análisis funcional	Media	Ninguno	Ada	Resuelta
TBOARQ-2985 [Backend] Implementación	Media	Ninguno	Ada	Resuelta
TBOARQ-2986 [Frontend] Implementación	Media	Ninguno	Ada	Resuelta
TBOARQ-2987 [Testing] Tests unitarios backend	Media	Ninguno	Ada	Resuelta
TBOARQ-2988 [Seguridad] Revisión backend	Media	Ninguno	Ada	Resuelta
TBOARQ-2989 [Seguridad] Revisión frontend	Media	Ninguno	Ada	Resuelta
TBOARQ-2990 [Validación] MR Backend	Media	Ninguno	Ada	Resuelta
TBOARQ-2991 [Validación] MR Frontend	Media	Ninguno	Ada	Resuelta
TBOARQ-2992 [QA E2E] Tests Playwright	Media	Ninguno	Ada	Resuelta
TBOARQ-2993 [Compliance] Auditoría ISO 27001	Media	Ninguno	Ada	Resuelta

Enlaces web

Análisis Funcional	Fecha de actualización: hace 20 minutos
MR frontend: feat(DID-16906)	
Compliance Report	Fecha de actualización: hace 3 minutos



[DID-16906] AF — DEMO REDHAT SUMMIT

De Ada  Escuchar  Añade una reacción

Análisis Funcional — DEMO REDHAT SUMMIT

Requisito original

> Quiero que generes una pantalla donde se vea un camión de Carreras, el logo de RedHat y abajo ponga Carreras loves Redhat con un corazón y abajo Gracias Redhat Summit 2026

Fuente: [IN-7925](#) — DEMO REDHAT SUMMIT

Épica de ejecución: [DID-16906](#) (proyecto `itapp`, perfil `tbo`)

Adjuntos: ninguno (`.pipeline/DID-16906/attachments/` no existe).

Descomposición literal del enunciado — son cuatro elementos visuales en orden vertical:

#	Elemento pedido	Interpretación
1	«un camión de Carreras»	Imagen de camión con identidad Carreras, en la zona superior
2	«el logo de RedHat»	Logotipo Red Hat, junto/debajo del camión
3	«abajo ponga Carreras loves Redhat con un corazón»	Texto literal <code>Carreras loves Redhat</code> + un corazón
4	«abajo Gracias Redhat Summit 2026»	Texto literal <code>Gracias Redhat Summit 2026</code>



feat(DID-16906): añadir pantalla demo Redhat Summit 2026

EditCode⋮

 Open Ada requested to merge `feature/DID-16906` into `main` 11 minutes ago

Overview0Commits2Pipelines1Changes21



Resumen

Añade una pantalla estática de demo ("Redhat Summit 2026") al portal itapp, accesible vía `#/redhat-summit`. Sin backend, sin llamadas HTTP, sin inputs de usuario — pantalla puramente visual (SVGs inline + textos con Transloco).

Épica JIRA: [DID-16906](#)

Archivos Creados

- `src/app/views/redhat-summit/redhat-summit.component.ts`
- `src/app/views/redhat-summit/redhat-summit.component.html`
- `src/app/views/redhat-summit/redhat-summit.component.sass`
- `src/app/views/redhat-summit/redhat-summit.component.spec.ts`

Archivos Modificados

- `src/app/app.module.ts` — declaración del componente (NgModule clásico, `standalone: false`, convención real del repo)
- `src/app/app-routing.module.ts` — ruta eager `redhat-summit`, insertada antes del wildcard `**`
- `src/assets/i18n/es.json` — bloque `redhat-summit` (4 claves de texto)
- `package.json` — bump de versión `26.8.14` → `26.8.15`
- `CHANGELOG.md` — entrada `[26.8.15]`
- `docs/DID-16906/` — documentación del pipeline (analysis, reportes, codemaps)

Resultado de Validaciones

Validación	Resultado
Seguridad (secrets, innerHTML, bypass)	 PASS — sin hallazgos (ver <code>docs/DID-16906/security-frontend-report.md</code>)
na build --	 0 errores

0 Assignees

Edit

None - assign yourself

Reviewer

Edit

None - assign yourself

Labels

Edit

None

Milestone

Edit

None

Work items



None

Time tracking

 +

No estimate or time spent

1 Participant





RED HAT SUMMIT 2026



Red Hat

Carreras loves Red Hat

Gracias Red Hat Summit 2026





Connect

Thank you



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



x.com/RedHat