

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS



SI807 SISTEMAS DE INTELIGENCIA DE NEGOCIOS
“Aplicación de Sistemas de Inteligencia de Negocios en BBVA”

PRACTICA N°2

GRUPO N°4:

Código	Apellidos y Nombres	Correo Electrónico	Tareas Realizadas
20222029J	Cárdenas Palacios Leonardo Gustavo	leogcardenasp@gmail.com	Preguntas de negocio y definición de KPIs clave.
20222048D	Espinoza Cerna, Alex	espinozacernaalex@gmail.com	Diseño de modelo conceptual e inventario de fuentes de datos OLTP.
20222144C	Inocente Caro, Miguel Anderson	miguelander30@gmail.com	Ingesta de datasets en HDFS y creación de tablas en Hive. Consulta básica de control en Hive.

Profesores: Dr. Ing. Aradiel Castaneda, Hilario y García Atuncar, Fernando

Septiembre, 2025

Índice

- 1. DESARROLLO
 - 1.1 Preguntas de Negocio
 - 1.2. KPI's definidos
 - 1.3. Modelo conceptual preliminar
 - 1.4. Inventario de fuentes OLTP
- 2. EVIDENCIA TÉCNICA
 - 2.1. Implementación de Hortonworks
- 3. REFERENCIAS BIBLIOGRÁFICAS

Índice de Figuras

- Figura 1. Niveles de Certificación por % de Adopción del Practitioner
- Figura 2. Niveles de Certificación por % de Adopción del Continuous Integration
- Figura 3. Marco Playbook de los niveles de certificación
- Figura 4. Modelo Conceptual Estrella de la data Practitioner
- Figura 5. Modelo Conceptual Estrella de la data Continuous Integration

Índice de Tablas

- Tabla 1. Formulación de Preguntas de Negocio
- Tabla 2. Inventario de Fuentes OLTP

1. DESARROLLO

1.1. Preguntas del Negocio

La identificación de preguntas de negocio constituye un paso fundamental en la metodología Hefesto para el desarrollo de sistemas de Business Intelligence, ya que permite establecer el puente entre las necesidades organizacionales y los requerimientos técnicos del sistema de inteligencia de negocios. En el contexto de BBVA, estas preguntas emergen directamente de los problemas de negocio identificados previamente y se alinean con las necesidades de información de los distintos niveles organizacionales.

El proceso de certificación de servicios tecnológicos en BBVA, que comprende los niveles Practitioner y Continuous Integration, genera múltiples interrogantes que requieren respuestas basadas en datos para la toma de decisiones efectiva. Estas preguntas no solo reflejan la necesidad de monitorear el cumplimiento de los 17 KPIs definidos, sino que también abordan aspectos críticos como la seguridad, la calidad del desarrollo, la eficiencia operativa y la madurez organizacional.

La metodología aplicada para la definición de estas preguntas de negocio considera tres elementos clave:

- La identificación de los usuarios objetivo, principalmente Service Owners y gerentes del área de Engineering.
- La determinación del nivel de prioridad, basado en el impacto al negocio y el riesgo operativo.
- La vinculación directa con las fuentes de datos disponibles en el ecosistema tecnológico actual de BBVA.

En este sentido, las preguntas de negocio identificadas se centran principalmente en el área de *Engineering*, dado que es la responsable de la gestión y certificación de servicios tecnológicos según el modelo de madurez establecido. Esta focalización se justifica porque todos los indicadores definidos (desde fichas RFO hasta análisis de vulnerabilidades) son competencia directa de los equipos técnicos y sus responsables.

La priorización de las preguntas se realizó considerando criterios como el impacto directo en la continuidad operativa, el riesgo de seguridad, el cumplimiento regulatorio y la contribución a los objetivos estratégicos de transformación digital del banco. Así, las preguntas de alta prioridad abordan aspectos críticos que pueden afectar la reputación, la seguridad o la competitividad de BBVA, mientras que aquellas clasificadas como de prioridad media se enfocan en optimizaciones y mejoras incrementales que contribuyen al fortalecimiento progresivo del modelo de certificación y gestión de servicios tecnológicos.

Tabla 1. Formulación de Preguntas de Negocio

Área	Rol de Usuario	Pregunta de Negocio	Nivel de Prioridad	Fuente de Datos Actual
Engineering	Service Owner	¿Qué servicios N1 tienen mayor porcentaje de adopción del nivel Practitioner y cuáles requieren intervención inmediata?	Alta	Nucleus, Continuum, JIRA, Chimera, Bitbucket
Engineering	Service Owner	¿Cuáles son los servicios con mayor número de vulnerabilidades de alto riesgo y cómo ha evolucionado esta métrica mensualmente?	Alta	Chimera, Bitbucket, Nucleus
Engineering	Service Owner	¿Qué porcentaje de fichas RFO están en estado "OK" por servicio N1 y qué impacto tienen en los tiempos de puesta en producción?	Alta	Nucleus, Continuum
Engineering	Service Owner	¿Cuál es la calidad promedio de las features desplegadas por servicio y cómo se compara con el objetivo del 90%?	Media	JIRA
Engineering	Gerente de Área	¿Cuál es el nivel de madurez promedio de los servicios por UOL1 y UOL2 en los niveles Practitioner y Continuous Integration?	Alta	Nucleus, Continuum, JIRA, Chimera, Bitbucket
Engineering	Service Owner	¿Qué servicios N2 carecen de dependencias asignadas y qué riesgo operativo representan?	Media	Nucleus
Engineering	Service Owner	¿Cuáles el tiempo medio de integración de código y construcción de pipelines por servicio, y cómo impacta en la agilidad del desarrollo?	Media	Bitbucket, Jenkins
Engineering	Service Owner	¿Qué porcentaje de repositorios cumplen con la nomenclatura estándar y están gobernados en el análisis de seguridad?	Media	Bitbucket, Chimera, GLAM
Engineering	Gerente de Área	¿Cuáles son los principales cuellos de botella en el proceso de certificación y cómo afectan el time-to-market?	Alta	JIRA, Bitbucket, Jenkins, Nucleus, Continuum
Engineering	Service Owner	¿Qué historias de usuario carecen de pruebas de aceptación ejecutadas y qué riesgo de calidad representan?	Media	JIRA
Riesgos	Risk Manager	¿Cuáles la evolución mensual de vulnerabilidades por cada mil líneas de código y qué servicios representan mayor riesgo de seguridad?	Alta	Chimera, Bitbucket
Engineering	Gerente de Área	¿Qué unidades organizacionales (UOL) tienen menor adopción de prácticas ágiles y requieren planes de mejora prioritarios?	Alta	Todos los sistemas integrados
Engineering	Service Owner	¿Cuáles el porcentaje de construcciones correctas y tiempo medio de arreglo de fallos por servicio?	Media	Jenkins
Planeación Estratégica	Directorio/ Gerencia General	¿Cuáles el nivel de transformación digital actual medido por los indicadores de madurez y cómo se compara con los objetivos estratégicos?	Alta	Consolidado de todos los sistemas
Engineering	Service Owner	¿Qué servicios tienen análisis técnicos pendientes de revisión por más de 7 días y qué impacto tienen en la planificación?	Media	JIRA

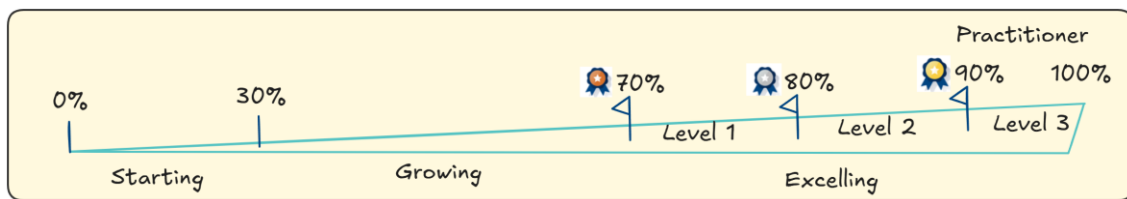
Fuente: Elaboración propia.

1.2. KPI's definidos

Para definir los KPI's se necesita primero definir que son los niveles de madurez Practitioner y Continuous Integration.

Practitioner: El servicio adopta prácticas básicas que permiten iniciar su camino de madurez, estableciendo una base sólida para niveles superiores.

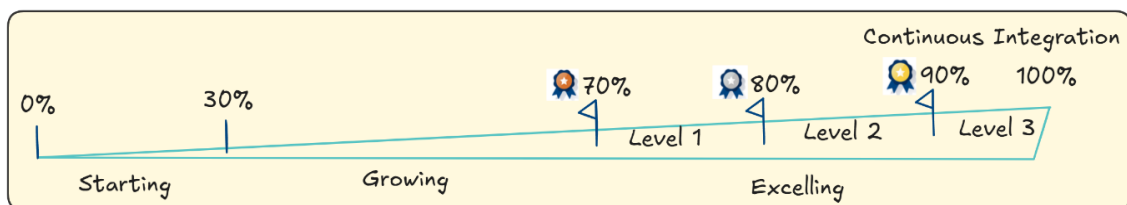
Figura 1. Niveles de Certificación por % de Adopción del Practitioner



Fuente: *Elaboración Propia.*

Continuous Integration: El servicio tiene integrado su código de manera frecuente y automática en un repositorio compartido, asegurando consistencia, detección temprana de errores y confiabilidad en el proceso de desarrollo.

Figura 2. Niveles de Certificación por % de Adopción del Continuous Integration



Fuente: *Elaboración Propia.*

Ambos niveles de madurez de estos servicios nos ayudan a tener una vista clara del SDLC (Software Development LifeCycle).

I. Indicadores para alcanzar el nivel Practitioner

A) Inventario: Agrupación en la que se miden indicadores relacionados con el inventario de servicios tecnológicos de Nucleus y de las fichas Ready For Operation (RFO) de estos servicios en Continuum.

a) Fichas de RFO para Servicios N2 de cada Servicio N1 con estados considerados «OK»

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Fichas de RFO para Servicios N2 de cada Servicio N1 con estados considerados «OK»
Objetivo estratégico asociado	<ul style="list-style-type: none"> Conocer / asegurar el mantenimiento del inventario tecnológico. Definir y asegurar (cuando aplique) la estrategia de recuperación acorde con la criticidad del servicio. Asegurar la monitorización técnica del estado del servicio.
Definición	Porcentaje de fichas de RFO para Servicios N2 de cada Servicio N1 cuyo estado se considera como «OK» para continuar subiendo a producción.
Fórmula	$\frac{\# \text{ de fichas de Servicios N2 del SN1 con RFO con status «OK»}}{\# \text{ de fichas de Servicios N2 del SN1}} \times 100\%$ <p>* Si hay una división entre cero en la fórmula, no se considera el indicador.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Nucleus, Continuum
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	13% (indicador de peso bajo)

b) Servicios N2 del servicio N1 con dependencias asignadas

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Servicios N2 del servicio N1 con dependencias asignadas
Objetivo estratégico asociado	Conocer y asegurar el mantenimiento del inventario tecnológico del servicio y fichas RFO.
Definición	Porcentaje de Servicios N2 de cada Servicio N1 con dependencias (aquellas en las que se establece de qué otros servicios dependen el servicio analizado) asignadas en Nucleus.
Fórmula	$\frac{\# \text{ de Servicios N2 del Servicio N1 con dependencias}}{\# \text{ de Servicios N2 del Servicio N1}} \times 100\%$ <p>* Si hay una división entre cero en la fórmula, no se considera el indicador.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Nucleus
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	13% (indicador de peso bajo)

B) Modelo Operativo: Incluye un sólo indicador que le da mayor responsabilidad al Service Owner sobre el % de Adopción de cada uno de sus Servicios N2 de los Playbooks de Desarrollo.

a) Cumplimiento del objetivo de adopción del nivel (en función de la priorización del SN2) para los Servicios N2 del Servicio N1

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Cumplimiento del objetivo de adopción para los Servicios N2 del Servicio N1.
Objetivo estratégico asociado	Supervisar el proceso de desarrollo de nuevas funcionalidades y participar en resolución de impedimentos.
Definición	Porcentaje de Servicios N2 de cada Servicio N1 que cumplen con el objetivo de adopción.
Fórmula	$\frac{\text{Puntaje total de adopción de SN2}}{\text{Total de SN2 medidos del SN1}} \times 100\%$ * Si hay una división entre cero en la fórmula, no se considera el indicador.
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Chimera, Nucleus
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	27% (indicador de peso bajo)

C) Features y Desarrollos: Se cuenta con 1 indicador que permite medir la calidad de las features que se despliegan / llevan en los servicios.

a) Calidad de las Features de los Servicios N2 de cada Servicio N1

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Calidad de las Features de los Servicios N2 de cada Servicio N1
Objetivo estratégico asociado	<ul style="list-style-type: none"> Tener visión global del backlog de servicio y dar visibilidad a negocio. Supervisar el proceso de desarrollo de nuevas funcionalidades y participar en resolución de impedimentos.
Definición	Porcentaje de Features del SN2 de cada SN1 que cumplen con los criterios de calidad definidos.
Fórmula	$\frac{\text{Features desplegadas que cumplen con los criterios de Calidad}}{\text{Total de features desplegadas consideradas}} \times 100\%$ * Si hay una división entre cero en la fórmula, no se considera el indicador.
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	JIRA
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	20 % (indicador de peso medio)

D) Seguridad: Se ha definido un indicador cuyo objetivo es medir la seguridad del código del servicio ante las vulnerabilidades.

a) Evolución de vulnerabilidades de alto riesgo por cada mil líneas de código en las UUAA's asociadas al servicio

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Evolución de vulnerabilidades de alto riesgo por cada mil líneas de código en las UUAA's asociadas al servicio
Objetivo estratégico asociado	<ul style="list-style-type: none"> Garantizar el cumplimiento de regulaciones y normativas. Garantizar la seguridad del servicio ante vulnerabilidades.
Definición	Mide la variación porcentual de vulnerabilidades de alto riesgo detectadas por cada 1000 líneas de código en los repositorios asociados a un servicio (analizados en Bitbucket y mediante Chimera), comparando el resultado con el dato del mes anterior.
Fórmula	<ul style="list-style-type: none"> Vulnerabilidad (cada mil líneas de código): $\frac{\text{Total de vulnerabilidades high}}{\text{Total de líneas de código}} \times 1000$ Evolución de Vulnerabilidades: $\frac{\text{Vulnerabilidad del mes actual} - \text{Vulnerabilidad del mes anterior}}{\text{dato vulnerabilidad del mes anterior}} \times 100\%$ <p>* Si no se desarrolló ninguna línea de código en el mes actual, no se considera el indicador. * Si hay una división entre cero en la fórmula de vulnerabilidades, se considera el valor de -99%. (para el mes anterior) * Si en la evolución de vulnerabilidades sale -100% se considera como 0%.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Nucleus, Quimera, BitBuckect
Responsable	Service Owner en el área de Engineering
Meta	100 %
Umbrales (Semáforos)	<ul style="list-style-type: none"> Umbral Central: Vulnerabilidad (actual) < 0.04 → 100% adopción. Vulnerabilidad (actual) > 0.2 → 0% adopción. Si el resultado se encuentra fuera del umbral central se utiliza la segunda fórmula (Evolución de vulnerabilidades) que combina la Vulnerabilidad actual y del mes anterior, y sigue la siguiente regla de umbral: Evol. Vulnerabilidad ≤ -10% → 100% adopción. Evol. Vulnerabilidad > -10% → 75% adopción. Evol. Vulnerabilidad = 0% → 50% adopción. Evol. Vulnerabilidad > 0% y Evol. Vulnerabilidad < 10% → 25 % adopción. Evol. Vulnerabilidad ≥ 10% → 0% adopción.
Peso del Indicador en la adopción	27% (indicador de peso alto)

* UUAA's: Unidades de Arquitectura y Aplicación.

E) Adopción y Certificación: Se descargan datos de distintas fuentes (Nucleus, Continuum, JIRA, Chimera) en un repositorio global. Luego se aplican las fórmulas y pesos definidos para cada indicador, redistribuyendo si alguno no aplica, y al final se obtiene el porcentaje de adopción.

a) % de adopción total del practitioner

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	% de adopción total del practitioner
Objetivo estratégico asociado	Impulsar la adopción del nivel de practitioner en los servicios del banco para garantizar estandarización, eficiencia y madurez operativa.
Definición	Mide el porcentaje de adopción del practitioner en los servicios N1 y N2, considerando los distintos indicadores y niveles de certificación.
Fórmula	$\text{Adopción (\%)} = \frac{\{[(\text{Peso del indicador 1}) \times (\text{Resultado del indicador 1})] + [(\text{Peso del indicador 2}) \times (\text{Resultado del indicador 2})] + [(\text{Peso del indicador } \dots) \times (\text{Resultado del indicador } \dots)] + [(\text{Peso del indicador 5}) \times (\text{Resultado del indicador 5})]\}}{\{\text{Suma de los pesos de los indicadores considerados} \}}$ <p>* Si no se consideran algunos indicadores en un servicio, en vez de dividir entre 1 al Adopción (%), se dividirá entre la suma de los pesos de los indicadores que se consideran.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Resultado de adopción de cada indicador.
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Level 3: ≥ 90 % Level 2: 89-80 % Level 1: 79-70 % No Certificado: < 70 %

II. Indicadores para alcanzar el nivel Continuous Integration

A) Análisis y Diseño

a) % Análisis en estado «Analysis in Review» menor o igual a 7 días

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	% Análisis en estado «Analysis in Review» menor o igual a 7 días
Objetivo estratégico asociado	Garantizar la finalización oportuna de los análisis manteniendo el foco en el estado «Analysis in Review», para evitar retrasos y minimizar el impacto en equipos dependientes.
Definición	Porcentaje de análisis (de cualquier tipo) que han estado pendientes de revisión (con estado «Analysis in Review») durante 7 días o menos.
Fórmula	$\frac{\text{Issues Analysis que pasaron por «Analysis in Review» menor a 7 días}}{\text{Total de Issues Analysis que pasaron por «Analysis in Review»}} \times 100\%$

	* Si hay una división entre cero en la fórmula, no se considera el indicador.
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	JIRA
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	5% (indicador de peso bajo)

B) Gestión del Backlog

a) % Historias de usuario con Release/Fix Versión asociado

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	% Historias de usuario con Release/Fix Versión asociado
Objetivo estratégico asociado	Fomentar el uso de <i>releases</i> en Jira para asegurar la planificación, trazabilidad y visibilidad de los despliegues en producción, alineando los requerimientos de usuario con el código desarrollado.
Definición	Porcentaje de historias de usuario en estado « <i>Deployed</i> » durante el mes de medición que tienen el campo « <i>Fix Version</i> » informado.
Fórmula	$\frac{\text{Historias deployed con Fix Version}}{\text{Total de historias deployed}} \times 100\%$ <p>* Si hay una división entre cero en la fórmula, no se considera el indicador.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	JIRA
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	6% (indicador de peso bajo)

C) Desarrollo y Versionado de Código

a) % Repositorios con nomenclatura estándar

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	% Repositorios con nomenclatura estándar
Objetivo estratégico asociado	Estandarizar la nomenclatura de ramas en los repositorios para mejorar la colaboración entre equipos, facilitar el onboarding de nuevos desarrolladores, optimizar los flujos de trabajo en CI/CD y reducir errores derivados de la falta de consistencia en los nombres.
Definición	Porcentaje de repositorios activos que cumplen con la nomenclatura establecida en las ramas. Este indicador busca verificar la adopción

	progresiva del estándar, inicialmente de forma opcional y posteriormente obligatorio a través de la herramienta.
Fórmula	$\frac{\text{Repositorios activos con ramas que cumplen la nomenclatura estándar}}{\text{Total de repositorios activos}} \times 100\%$ <p>* Si hay una división entre cero en la fórmula, no se considera el indicador.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Bitbucket
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	6% (indicador de peso bajo)

b) Tiempo medio de aprobación de Pull Requests

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Tiempo medio de aprobación de Pull Requests
Objetivo estratégico asociado	Reducir los tiempos de aprobación de Pull Requests para agilizar el ciclo de desarrollo, minimizar bloqueos y fomentar buenas prácticas de revisión que mejoren la calidad del código y la eficiencia colaborativa de los equipos.
Definición	Tiempo de aprobación de Pull Requests (PR) mide cuánto tiempo pasa desde que se abre un PR hasta que es aceptada o rechazada por los revisores.
Fórmula	Mediana (tiempos en que los PR (Pull Request) han tardado en ser mergeadas o rechazadas)
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Bitbucket
Responsable	Service Owner en el área de Engineering
Meta	90 %
Umbrales (Semáforos)	Tm. Aprobación < 1 día → 90% Tm. Aprobación ≥ 1 día y < 1,5 días → 70% Tm. Aprobación ≥ 1,5 días y < 3 días → 30% Tm. Aprobación ≥ 3 días → 0%
Peso del Indicador en la adopción	10% (indicador de peso medio)

c) Tamaño Medio de Pull Requests

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Tamaño Medio de Pull Requests
Objetivo estratégico asociado	Mantener Pull Requests pequeños facilita revisiones más rápidas y efectivas, reduce tiempos de aprobación, minimiza errores en la integración y fomenta ciclos de feedback ágiles. Esto mejora la estabilidad del código, acelera la entrega continua y refuerza la motivación del equipo al ver sus cambios aprobados e integrados con mayor rapidez.

Definición	Para fomentar integraciones pequeñas y frecuentes este indicador mide el tamaño de las integraciones (Pull Request)
Fórmula	Mediana (líneas de código modificadas en PRs mergeadas o rechazadas a ramas permanentes)
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Bitbucket
Responsable	Service Owner en el área de Engineering
Meta	90 %
Umbral (Semáforos)	Tm. PR < 300 líneas de código → 90% Tm. PR ≥ 300 y < 400 líneas de código → 70% Tm. PR ≥ 440 y < 600 líneas de código → 30% Tm. PR ≥ 600 líneas de código → 0%
Peso del Indicador en la adopción	5% (indicador de peso bajo)

d) Repositorios gobernados en el análisis estático de Seguridad

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Repositorios gobernados en el análisis estático de Seguridad
Objetivo estratégico asociado	Asegurar que los repositorios integrados en Chimera gestionen activamente sus vulnerabilidades, garantizando el compromiso de los equipos de desarrollo con la seguridad y minimizando el riesgo de acumulación de fallas críticas en el código.
Definición	Este indicador mide los repositorios gobernados en el análisis estático de seguridad (Chimera). Para que un repositorio esté «gobernado», no sólo es necesario que esté integrado, sino que también los desarrolladores tengan acceso a Chimera y exista gestión activa de las debilidades de seguridad detectadas por la herramienta.
Fórmula	$\frac{\text{Repositorios activos gobernados en Chimera}}{\text{Total de repositorios activos}} \times 100\%$
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Chimera, GIAM, Bitbucket
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbral (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	7% (indicador de peso medio)

D) Análisis y construcción de código

a) Tiempo medio de integración

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Tiempo medio de integración
Objetivo estratégico asociado	<ul style="list-style-type: none"> Identificar cuellos de botella en revisiones, pruebas, pipelines o dependencias que ralentizan la integración.

	<ul style="list-style-type: none"> Promover buenas prácticas al reflejar la salud del pipeline y fomentar commits pequeños y revisiones rápidas. Acelerar el ciclo de desarrollo entregando valor más rápido, mejorando la calidad del software y motivando al equipo.
Definición	Tiempo desde que se hace una primera contribución a una rama hasta que se integra a rama develop, release o master/main.
Fórmula	Mediana (tiempo desde 1era contribución hasta el merge en ramas permanentes)
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Bitbucket
Responsable	Service Owner en el área de Engineering
Meta	90 %
Umbral (Semáforos)	Tm. Integración < 3 días → 90% Tm. Integración ≥ 3 días y < 4 días → 70% Tm. Integración ≥ 4 días y < 6 días → 30% Tm. Integración ≥ 6 días → 0%
Peso del Indicador en la adopción	16% (Indicador de peso alto)

b) Tiempo medio construcciones

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Tiempo medio construcciones
Objetivo estratégico asociado	<ul style="list-style-type: none"> Reducir tiempos de espera para integrar cambios fluidamente, mantener al equipo productivo y acelerar el ciclo de desarrollo. Identificar cuellos de botella en pruebas o compilaciones, fomentando buenas prácticas y mejorando la eficiencia del equipo. Detectar anomalías en los tiempos de construcción para corregir fallos rápido y asegurar un desarrollo estable y de calidad.
Definición	Mide el tiempo promedio que tardan los procesos automáticos (pipelines) en completarse. Aplica a todas las ramas y todas las compilaciones, ya que el desarrollador se ve impactado en el tiempo de ejecución de los pipelines en todas las ramas.
Fórmula	Mediana (tiempo de ejecución de pipelines correctos en cada repositorio)
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Jenkins
Responsable	Service Owner en el área de Engineering
Meta	90 %
Umbral (Semáforos)	Tm. Construcciones < 20 min → 90% Tm. Construcciones ≥ 20 min y < 35 min → 70% Tm. Construcciones ≥ 35 min y < 45 min → 30% Tm. Construcciones ≥ 45 min → 0%
Peso del Indicador en la adopción	9% (Indicador de peso medio)

c) % Construcciones correctas

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	% Construcciones correctas
Objetivo estratégico asociado	<ul style="list-style-type: none"> Fomentar la responsabilidad del desarrollador aplicando buenas prácticas y asegurando calidad desde el primer commit. Contribuir a un flujo de trabajo más fluido reduciendo errores, retrocesos y demoras en la entrega. Incrementar la confianza en CI/CD, fortalecer la colaboración y promover la mejora continua del equipo.
Definición	Mide el porcentaje de pipelines que finalizan exitosamente en cualquier rama, reflejando si los cambios subidos por los desarrolladores cumplen con los criterios básicos de calidad, como la compilación correcta del código, el paso exitoso de pruebas unitarias, y la ejecución de etapas adicionales.
Fórmula	$\frac{\text{Ejecuciones de pipeline con estado SUCCESS}}{\text{Ejecuciones de pipeline con estado SUCCESS o FAILURE}} \times 100\%$ <p>* Si hay una división entre cero en la fórmula, no se considera el indicador.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Jenkins
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	13% (Indicador de peso alto)

d) Tiempo medio en arreglar construcciones

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Tiempo medio en arreglar construcciones
Objetivo estratégico asociado	<ul style="list-style-type: none"> Promover la cultura de calidad para prevenir fallos, resolverlos rápido y fomentar la mejora continua. Reducir el tiempo de bloqueo para evitar interrupciones, efectos en cascada y retrasos en despliegues. Facilitar la detección de patrones recurrentes que permitan mejorar procesos y elevar la moral del equipo.
Definición	Mide el tiempo promedio que tarda un equipo en corregir una construcción fallida desde el momento en que ocurre el fallo hasta que se ejecuta y completa exitosamente el pipeline correspondiente.
Fórmula	Mediana (tiempo entre fallo y éxito del pipeline en ramas permanentes)
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Jenkins
Responsable	Service Owner en el área de Engineering

Meta	90 %
Umbrales (Semáforos)	Tm. Arreglo < 60 min → 90% Tm. Arreglo ≥ 60 min y < 120 min → 70% Tm. Arreglo ≥ 120 min y < 180 min → 30% Tm. Arreglo ≥ 180 min → 0%
Peso del Indicador en la adopción	8% (Indicador de peso medio)

E) Testing

a) Calidad del código

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	Calidad del código
Objetivo estratégico asociado	Asegurar que todo el código nuevo desarrollado cumple con los estándares de la industria, de manera que no se incremente la deuda técnica y se mejore progresivamente la mantenibilidad del código.
Definición	Porcentaje de repositorios con actividad en Sonar cuyas ramas develop, master, main, release y release cumplen con los criterios de calidad y cobertura de código definidos en SonarQube.
Fórmula	$\frac{\text{Repositorios con actividad y con análisis SonarQube OK}}{\text{Repositorios con actividad y con análisis SonarQube}} \times 100\%$ <p>* Si hay una división entre cero en la fórmula, no se considera el indicador.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	SonarQube
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	10% (Indicador de peso medio)

b) % Historias de Usuario, Dependencias y Bugs con pruebas de aceptación (XRay)

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	% Historias de Usuario, Dependencias y Bugs con pruebas de aceptación (XRay)
Objetivo estratégico asociado	<ul style="list-style-type: none"> Mejorar el alineamiento entre las necesidades del cliente y los incrementos funcionales que el equipo desarrolla. Auditar las evidencias de ejecución de las pruebas de aceptación gracias a que se mide la cobertura funcional y su correcta ejecución.
Definición	Porcentaje de Historias de Usuario, Dependencias y Bugs de JIRA desplegados de forma planificada, que tienen resultados de ejecución de pruebas de aceptación en JIRA XRay (Test Execution con resultado final «pass» o «fail»).
Fórmula	

	$\frac{\text{Ítems desplegados con pruebas Xray ejecutadas}}{\text{Ítems desplegados}} \times 100\%$ <p>* Si hay una división entre cero en la fórmula, no se considera el indicador.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	JIRA
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Verde: ≥ 90 %, Amarillo: 89-70%, Rojo: < 69 %
Peso del Indicador en la adopción	5% (Indicador de peso bajo)

E) Adopción y Certificación:

a) % de adopción total del continuous integration

CAMPO	DESCRIPCIÓN / EJEMPLO
Nombre del KPI	% de adopción total del continuous integration
Objetivo estratégico asociado	Impulsar la adopción del nivel de continuous integration en los servicios del banco para garantizar estandarización, eficiencia y madurez operativa.
Definición	Mide el porcentaje de adopción del continuous integration en los servicios N1 y N2, considerando los distintos indicadores y niveles de certificación.
Fórmula	$\text{Adopción (\%)} = \frac{\{[(\text{Peso del indicador 1}) \times (\text{Resultado del indicador 1})] + [(\text{Peso del indicador 2}) \times (\text{Resultado del indicador 2})] + [(\text{Peso del indicador } \dots) \times (\text{Resultado del indicador } \dots)] + [(\text{Peso del indicador 12}) \times (\text{Resultado del indicador 12})]\}}{\{\text{Suma de los pesos de los indicadores considerados} \}}$ <p>* Si no se consideran algunos indicadores en un servicio, en vez de dividir entre 1 al Adopción (%), se dividirá entre la suma de los pesos de los indicadores que se consideran.</p>
Unidad de Medida	%
Frecuencia de Medición	Mensual
Fuente de Datos	Resultado de adopción de cada indicador.
Responsable	Service Owner en el área de Engineering
Meta	≥ 90 %
Umbrales (Semáforos)	Level 3: ≥ 90 % Level 2: 89-80% Level 1: 79-70% No Certificado: < 70%

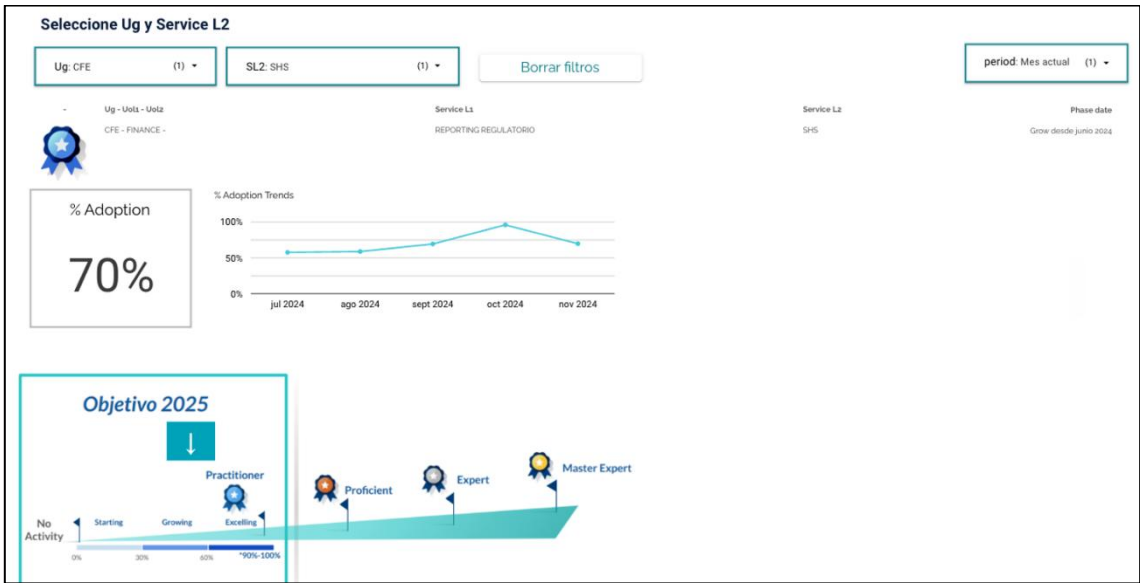
El Marco Playbook de BBVA es una guía estructurada que define los niveles progresivos de madurez tecnológica que deben alcanzar los servicios digitales: Starting, Growing, Excelling y un cuarto nivel variable (como Practitioner, Continuous Integration, Continuous Continuous Delivery, Continuous Deployment), el cual representa el objetivo específico de certificación de cada servicio. Este marco se basa en el cumplimiento de un conjunto de los KPIs detallados previamente.

Actualmente, el Service Owner realiza de forma manual el cálculo y seguimiento de estos indicadores, consolidando información dispersa en múltiples fuentes (Jira, Bitbucket, Nucleus, Chimera, Jenkins, entre otras) para determinar el porcentaje de adopción y el nivel de certificación alcanzado.

La Figura 3 representa precisamente ese resumen del estado actual de certificación, elaborado a partir de dicho análisis manual.

El propósito de este trabajo es automatizar ese proceso: mediante un sistema de Inteligencia de Negocios, se busca extraer, transformar y cargar los datos crudos de las fuentes originales para calcular los KPIs de forma automática y presentarlos en un dashboard dinámico, que permita al Service Owner monitorear en tiempo cercano al real el avance de sus servicios a través del marco Playbook, sin intervención manual ni riesgo de errores.

Figura 3. Marco Playbook de los niveles de certificación



Nota: BBVA (2025)

1.3. Modelo conceptual preliminar

Con el fin de facilitar el análisis y la visualización de los indicadores de certificación de servicios tecnológicos en BBVA, se han definido dos modelos conceptuales preliminares basados en una arquitectura de tipo estrella (star schema). Estos modelos permiten soportar el cálculo y monitoreo de los KPIs asociados a cada nivel de madurez: Practitioner y Continuous Integration.

A. Nivel 1: Mediciones del nivel Practitioner

a. Tabla de hechos: **fact_mediciones_practitioner**

Almacena los registros necesarios para el cálculo de los KPIs correspondientes a este nivel. Entre ellos destacan: número de fichas RFO correctas, número de dependencias asignadas, cantidad de features desplegadas, total de vulnerabilidades detectadas, entre otros.

b. Dimensiones:

- **dim_geográfica:** Posibilita el filtrado de resultados por región o país en el que opera el servicio.
- **dim_fecha:** Habilita el análisis temporal (mes, trimestre, año) para la identificación de tendencias.
- **dim_servicio_n1:** Identifica el servicio de nivel 1 (servicio padre) al que pertenecen los servicios de nivel 2 evaluados.

B. Nivel 2: Mediciones del nivel Continuous Integration

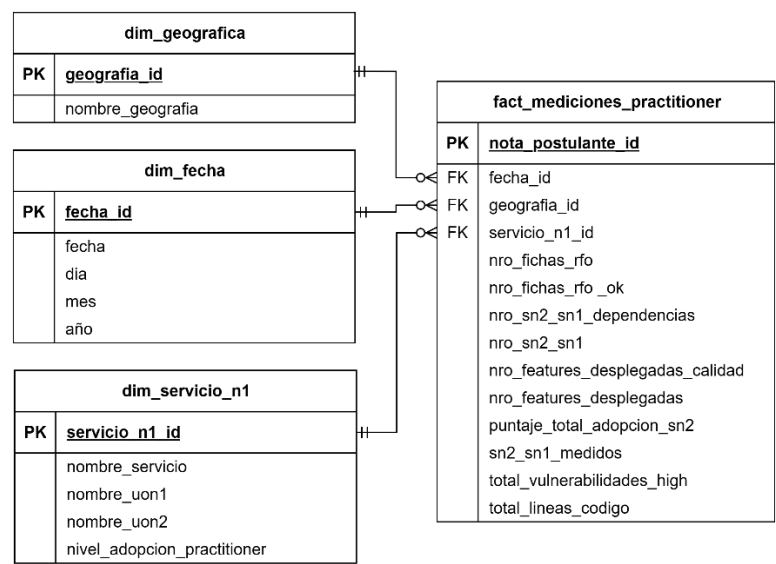
a. Tabla de hechos: **fact_mediciones_continuous_integration**

Contiene los registros que permiten calcular los 13 KPIs definidos para este nivel, tales como: tiempo de aprobación de pull requests, número de repositorios con nomenclatura estándar, tiempo medio de integración, tiempo promedio de construcción, entre otros.

b. Dimensiones:

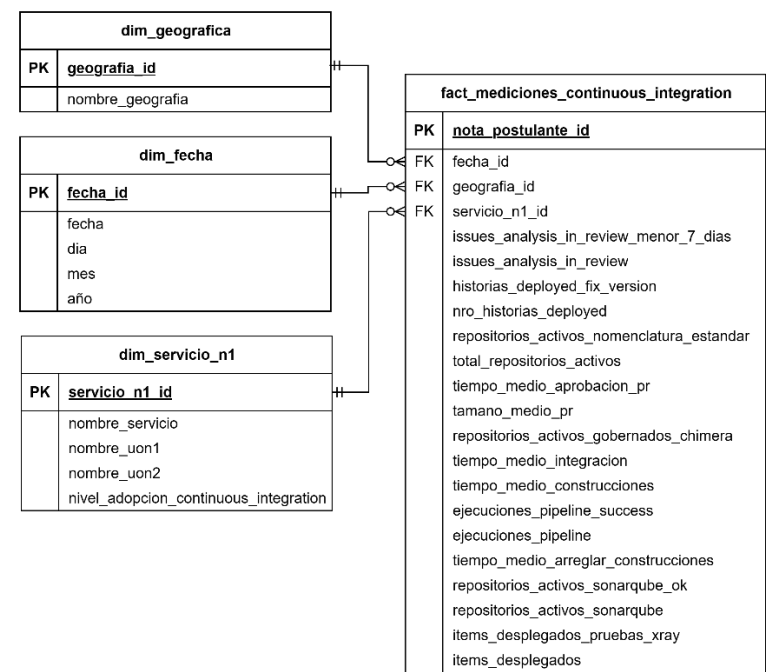
- **dim_geografica:** Idéntica al modelo anterior, para análisis por ubicación.
- **dim_fecha:** Para análisis temporal de métricas de CI/CD.
- **dim_servicio_n1:** Para agrupar y comparar resultados por servicio principal.

Figura 4. Modelo Conceptual Estrella de la data Practitioner



Fuente: *Elaboración propia.*

Figura 5. Modelo Conceptual Estrella de la data Continuous Integration



Fuente: *Elaboración propia.*

1.4. Inventario de fuentes OLTP

Inventario de fuentes OLTP tiene como propósito identificar y documentar los sistemas transaccionales internos y externos que generan los datos crudos necesarios para medir los KPIs de madurez tecnológica en BBVA, específicamente los niveles Practitioner y Continuous Integration. Este inventario es fundamental para entender desde dónde provienen los datos como Jira, Bitbucket, Nucleus, Continuum, Chimera, Jenkins, SonarQube y GIAM, quién los utiliza (área usuaria: Engineering, Security, DevOps), qué tipo de sistema son (transaccional, maestro, operativo), qué tecnología subyacente emplean (SaaS, plataforma interna, herramientas open source) y con qué frecuencia se actualizan (en tiempo real, diaria, mensual). Además, incluye observaciones clave sobre el rol de cada sistema en la trazabilidad, seguridad, calidad del código y gestión de servicios, lo que permite a los equipos de inteligencia de negocios diseñar una arquitectura de datos robusta, automatizada y escalable que transforme estos datos dispersos en información estratégica para la toma de decisiones.

Tabla 2. Inventario de Fuentes OLTP

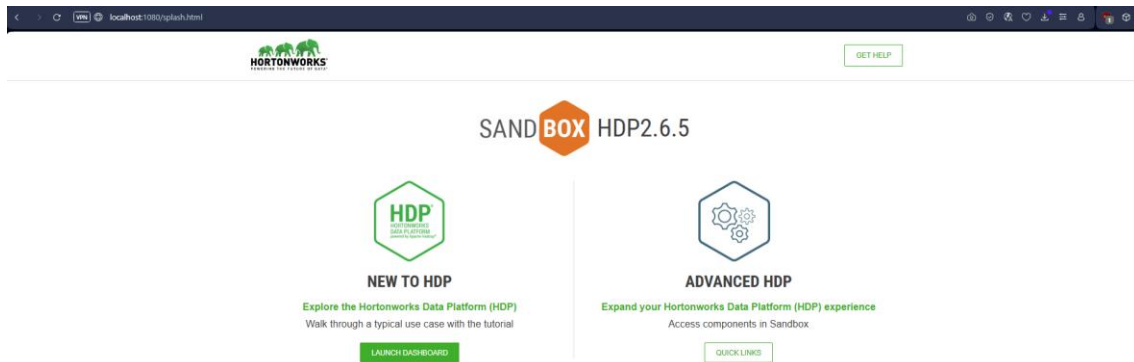
Sistema	Área usuaria	Tipo	Tecnología	Frecuencia actualización	Obs.
Jira	Engineering/ Service Owners	Transaccional/ Operativo	Plataforma SaaS (Atlassian)	En tiempo real	Fuente principal de historias, bugs, análisis, releases y trazabilidad de features.
Bitbucket	Engineering/ Desarrollo	Transaccional/ Operativo	Plataforma SaaS (Atlassian)	En tiempo real	Almacena código, PRs, commits; esencial para medir integración continua y calidad.
Nucleus	Engineering/ Gestión de Servicios	Maestro / Operativo	Plataforma interna BBVA	Diaria / Semanal	Sistema central de inventario tecnológico (servicios N1/N2, dependencias).
Continuum	Engineering/ Operaciones	Transaccional/ Operativo	Plataforma interna BBVA	Diaria	Gestiona fichas RFO (Ready For Operation) para certificación de servicios.
Chimera	Security / Engineering	Transaccional/ Operativo	Plataforma interna BBVA (SAST)	Diaria / Semanal	Análisis estático de seguridad (vulnerabilidades en código).
SonarQube	Quality / Engineering	Transaccional/ Operativo	Plataforma SaaS / On-premise	Diaria	Evalúa calidad técnica del código (bugs, cobertura, deuda técnica).
Jenkins	DevOps / Engineering	Transaccional/ Operativo	Plataforma CI/CD	En tiempo real	Ejecuta pipelines, construcciones y pruebas automáticas.
GIAM	Seguridad / IT	Maestro / Operativo	Plataforma interna BBVA	Semanal	Sistema de gestión de identidades y accesos a herramientas como Chimera.
Samuel	Engineering/ Gestión de Datos	Maestro / Operativo	Plataforma interna BBVA	Mensual	Posiblemente complemento o alias de Nucleus/Continuum para metadatos de servicios.

Fuente: *Elaboración Propia.*

2. EVIDENCIA TÉCNICA

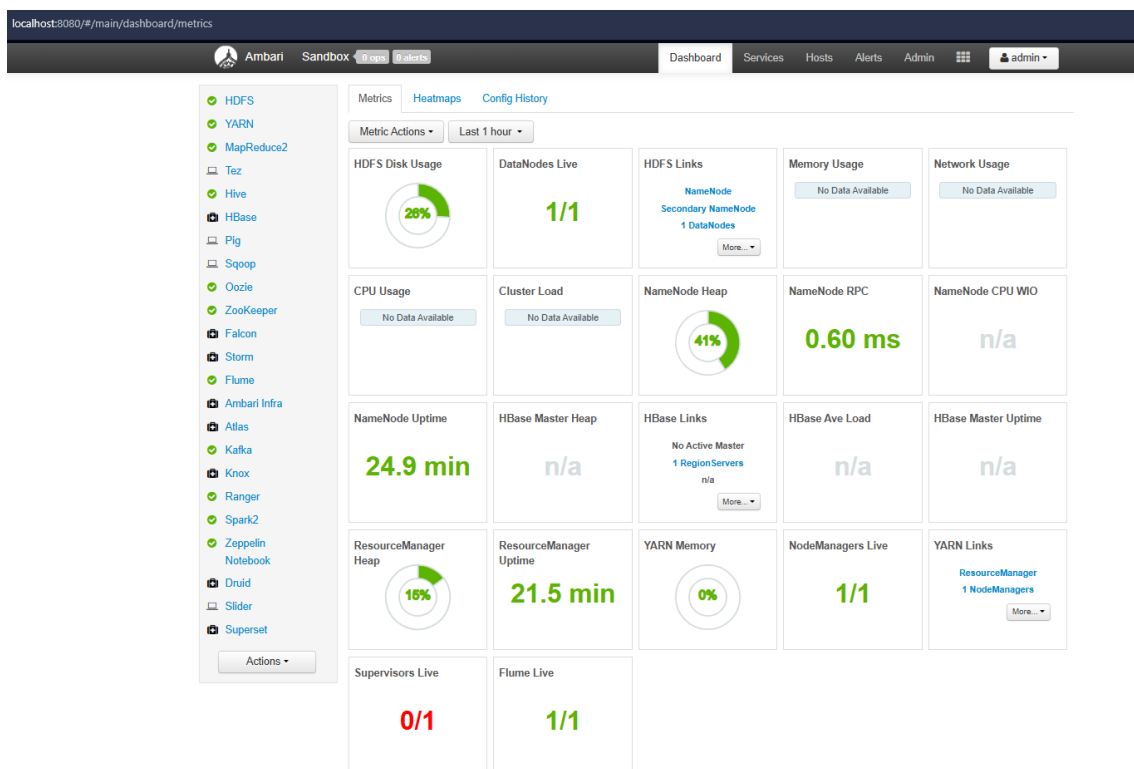
2.1. Implementación de Hortonworks

a) Entramos a Ambari desde la dirección <http://localhost:1080>.



Nota. *Interfaz SandBox*

b) Vemos el Dashboard de Ambari donde se visualiza los diferentes servicios en ejecución.



Nota. *Dashboard con todos los servicios de Ambari*

c) Modificamos la contraseña en Web Shell Client, el root pide cambiar la contraseña por primera vez y es s4ndb0x7 y luego cambiamos igual para el ambari

```
< > ↻ VPN localhost:4200
sandbox-hdp login: root
root@sandbox-hdp.hortonworks.com's password:
You are required to change your password immediately (root enforced)
Last login: Mon Sep 22 05:38:04 2025 from 172.18.0.2
Changing password for root.
(current) UNIX password:
New password:
Retype new password:
[root@sandbox-hdp ~]#
```

Nota. *Interfaz Web Shell Client cambiando la contraseña root*

```
< > ↻ VPN localhost:4200
sandbox-hdp login: root
root@sandbox-hdp.hortonworks.com's password:
Last login: Mon Sep 22 06:05:02 2025 from 172.18.0.2
[root@sandbox-hdp ~]# ambari-admin-password-reset
Please set the password for admin:
Please retype the password for admin:

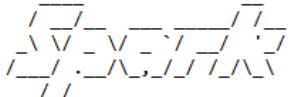
The admin password has been set.
Restarting ambari-server to make the password change effective...

Using python /usr/bin/python
Restarting ambari-server
Waiting for server stop...
Ambari Server stopped
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start.....
Server started listening on 8080

DB configs consistency check: no errors and warnings were found.
[root@sandbox-hdp ~]#
```

Nota. *Interfaz Web Shell Client cambiando la contraseña admin*

d) Ver la versión del Spark en Web Shell Client

```
< > ↺ VPN localhost:4200  
sandbox-hdp login: root  
root@sandbox-hdp.hortonworks.com's password:  
You are required to change your password immediately (root enforced)  
Last login: Mon Sep 22 05:38:04 2025 from 172.18.0.2  
Changing password for root.  
(current) UNIX password:  
New password:  
Retype new password:  
[root@sandbox-hdp ~]# spark-shell --version  
SPARK_MAJOR_VERSION is set to 2, using Spark2  
Welcome to  
 version 2.3.0.2.6.5.0-292  
  
Using Scala version 2.11.8, OpenJDK 64-Bit Server VM, 1.8.0_171  
Branch HEAD  
Compiled by user jenkins on 2018-05-11T08:28:14Z  
Revision b6b578c8dff89bfc07981ea3bda074262c3800ad  
Url git@github.com:hortonworks/spark2.git  
Type --help for more information.  
[root@sandbox-hdp ~]#
```

Nota. *Interfaz Web Shell Client visualizando la version del spark*

d) En HDFS colocamos los comandos para crear una carpeta y subir la data cruda relacionada al nivel de madurez Practitioner y Continuous Integration.

```

localhost:4200
sandbox-hdp login: root
root@sandbox-hdp.hortonworks.com's password:
Last login: Mon Sep 22 16:20:14 2025 from 172.18.0.2
[root@sandbox-hdp ~]# hdfs dfs -mkdir -p /user/root/data_cruda_practitioner
[root@sandbox-hdp ~]# hdfs dfs -mkdir -p /user/root/data_cruda_continuous_integration
[root@sandbox-hdp ~]#

```

Nota. *Interfaz Web Shell Client creando las carpetas*

The screenshot shows the Ambari web interface. At the top, there's a navigation bar with 'Ambari' and 'Sandbox' tabs. Below that, a breadcrumb trail shows 'user > root'. A yellow box indicates 'Total: 2 files or folders'. A search bar is present with the text 'Search in current directory...'. The main content is a table with the following columns: Name, Size, Last Modified, Owner, Group, and Permission. Two files are listed:

Name	Size	Last Modified	Owner	Group	Permission
data_cruda_continuous_integration	--	2025-09-22 11:39	root	hdfs	drwxr-xr-x
data_cruda_practitioner	--	2025-09-22 11:38	root	hdfs	drwxr-xr-x

Nota. *Interfaz Files View con las carpetas creadas*

Name	Size	Last Modified	Owner	Group	Permission
data_cruda_practitioner.csv	365.3 KB	2025-09-22 11:41	admin	hdfs	-rw-r--r--

Nota. *Interfaz Files View con la data cruda del practitioner subida*

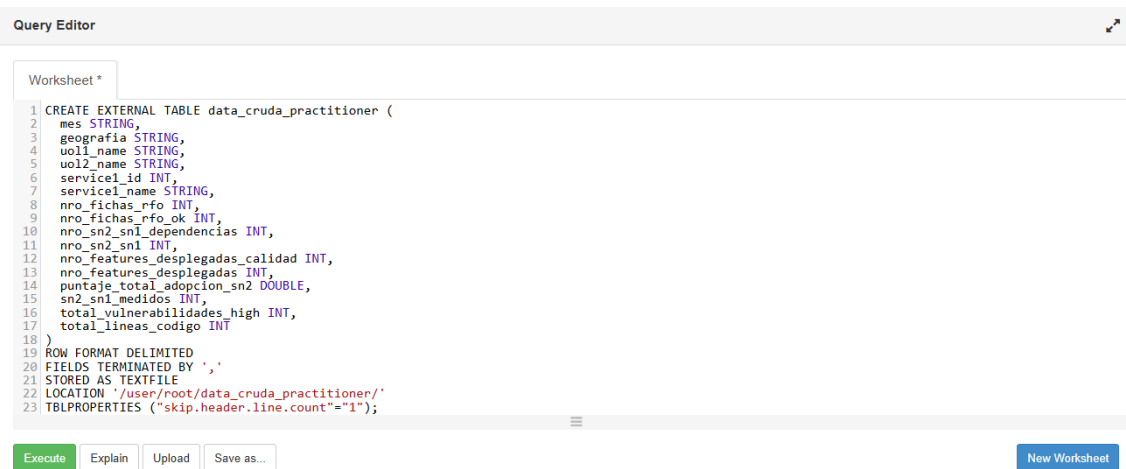
Name	Size	Last Modified	Owner	Group	Permission
data_cruda_continuous_integration.csv	943.4 KB	2025-09-22 11:41	admin	hdfs	-rw-r--r--

Nota. *Interfaz Files View con la data cruda del continuous integration subida*

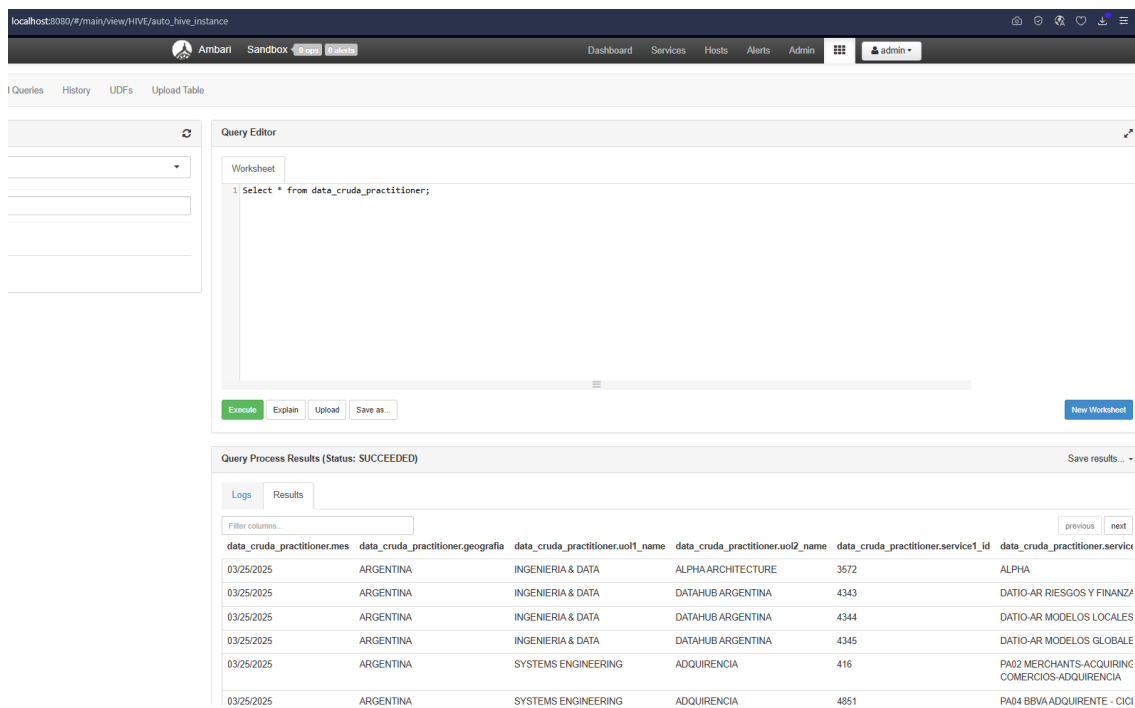
e) Conectar Hive con HDFS

- Para la data cruda del Practitioner

```
CREATE EXTERNAL TABLE data_cruda_practitioner (
  mes STRING,
  geografia STRING,
  uo11_name STRING,
  uo12_name STRING,
  service1_id INT,
  service1_name STRING,
  nro_fichas_rfo INT,
  nro_fichas_rfo_ok INT,
  nro_sn2_sn1_dependencias INT,
  nro_sn2_sn1 INT,
  nro_features_desplegadas_calidad INT,
  nro_features_desplegadas INT,
  puntaje_total_adopcion_sn2 DOUBLE,
  sn2_sn1_medidos INT,
  total_vulnerabilidades_high INT,
  total_lineas_codigo INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/user/root/data_cruda_practitioner/'
TBLPROPERTIES ("skip.header.line.count"="1");
```

Nota. *Interfaz Query Editor con la query de la tabla externa del practitioner*



Nota. *Interfaz Query Editor con la query para ver tabla externa del practitioner*

- Para la data cruda del Continuous Integration

```
CREATE EXTERNAL TABLE data_cruda_continuous_integration (  
  mes STRING,  
  geografia STRING,  
  uol1_name STRING,  
  uol2_name STRING,  
  sn1 INT,  
  issues_analysis_in_review_menor_7_dias INT,  
  issues_analysis_in_review INT,
```

```

historias_deployed_fix_version INT,
nro_historias_deployed INT,
repositorios_activos_nomenclatura_estandar DOUBLE,
total_repositorios_activos INT,
tiempo_medio_aprobacion_pr DOUBLE,
tamano_medio_pr DOUBLE,
repositorios_activos_gobernados_chimera INT,
tiempo_medio_integracion DOUBLE,
tiempo_medio_construcciones DOUBLE,
ejecuciones_pipeline_success INT,
ejecuciones_pipeline INT,
tiempo_medio_arreglar_construcciones DOUBLE,
repositorios_activos_sonarqube_ok INT,
repositorios_activos_sonarqube INT,
items_desplegados_pruebas_xray INT,
items_desplegados INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/user/root/data_cruda_continuous_integration/'
TBLPROPERTIES ("skip.header.line.count"="1");

```

The screenshot shows a web-based Query Editor interface. At the top, there's a title bar 'Query Editor' with a small icon. Below it, there are three tabs: 'Worksheet', 'Worksheet (2)', and 'Worksheet (3)'. The main area contains a SQL query for creating an external table. The query is as follows:

```

1 CREATE EXTERNAL TABLE data_cruda_continuous_integration (
2   mes STRING,
3   geografia STRING,
4   uo1_name STRING,
5   uo2_name STRING,
6   sn1 INT,
7   sn2 INT,
8   issues_analysis2_analysis_sin_review_menor_7_dias INT,
9   total_issues_analysis2_pasaron_analysis_sin_review INT,
10  historias_deployed_fix_version INT,
11  total_historias_deployed INT,
12  repositorios_activos_nomenclatura_estandar DOUBLE,
13  total_repositorios_activos INT,
14  mediana_tiempos_pr_tardado_ser_mergeadas_o_rechazadas DOUBLE,
15  mediana_lineas_codigo_modificadas_prs_mergeadas_o_rechazadas DOUBLE,
16  repositorios_activos_gobernados_chimera INT,
17  mediana_tiempo_desde_primera_contribucion_hasta_merge_permanentes DOUBLE,
18  mediana_tiempo_ejecucion_pipeline_correctos_repositorio DOUBLE,
19  ejecuciones_pipeline_estado_success INT,
20  ejecuciones_pipeline_estado_success_o_failure INT,
21  mediana_tiempo_entre_fallo_y_exitos_pipeline DOUBLE,
22  repositorios_actividad_y_analisis_sonarqube_ok INT,
23  repositorios_actividad_analisis_sonarqube INT,
24  items_desplegados_pruebas_xray_ejecutados INT,
25  items_desplegados INT
26 )
27 ROW FORMAT DELIMITED
28 FIELDS TERMINATED BY ','
29 STORED AS TEXTFILE
30 LOCATION '/user/root/data_cruda_continuous_integration/'
31 TBLPROPERTIES ("skip.header.line.count"="4");

```

At the bottom of the editor, there are buttons for 'Execute', 'Explain', 'Upload', and 'Save as...'. On the far right, there is a 'New Worksheet' button.

Nota. *Interfaz Query Editor con la query de la tabla externa del continuous integration*

The screenshot shows the Ambari Query Editor interface. The top navigation bar includes links for Dashboard, Services, Hosts, Alerts, Admin, and a user profile for 'admin'. The main area is divided into a left sidebar with 'Queries', 'History', 'UDFs', and 'Upload Table' options. The central 'Query Editor' pane contains a SQL query: `1 Select * from data_cruda_continuous_integration;`. Below the editor are buttons for 'Execute', 'Explain', 'Upload', and 'Save as...'. A 'New Worksheet' button is also present. The bottom section, 'Query Process Results (Status: SUCCEEDED)', shows a table of results with columns: `data_cruda_continuous_integration.mes`, `data_cruda_continuous_integration.geografia`, `data_cruda_continuous_integration.uo1_name`, `data_cruda_continuous_integration.uo2_name`, and `data_cruda_continuous_integration...`. The table contains 6 rows of data.

data_cruda_continuous_integration.mes	data_cruda_continuous_integration.geografia	data_cruda_continuous_integration.uo1_name	data_cruda_continuous_integration.uo2_name	data_cruda_continuous_integration...
01-Jul-25	ARGENTINA	SYSTEMS ENGINEERING	ADQUIRENCIA	2
01-Jul-25	ARGENTINA	SYSTEMS ENGINEERING	ADQUIRENCIA	81
01-Jul-25	ARGENTINA	SYSTEMS ENGINEERING	ADQUIRENCIA	82
01-Jul-25	ARGENTINA	SYSTEMS ENGINEERING	ADQUIRENCIA	83
01-Jul-25	ARGENTINA	SYSTEMS ENGINEERING	CANALES EMPRESAS	68
01-Jul-25	ARGENTINA	SYSTEMS ENGINEERING	CANALES EMPRESAS	69

Nota. *Interfaz Query Editor con la query para ver tabla externa del continuous integration*

f) Querys desde el Zeppelin

- Para la data cruda del Practitioner

➤ Mostrar las 5 primeras filas, ver el tipo de dato y conteo de filas.

```
%pyspark
csv_file = "hdfs:///user/root/data_cruda_practitioner/data_cruda_practitioner.csv"
# Lee el archivo CSV usando el API de Spark
df = spark.read.option("header", "true").option("inferSchema", "true").csv(csv_file)
```

```
%pyspark
# Muestra las primeras filas de la data
df.show(5)
```

```
%pyspark
# Muestra el esquema de la data (para verificar los tipos de datos inferidos)
df.printSchema()
```

```
%pyspark
# Muestra la cantidad de registros de la data
df.count()
```

Task 1: Leer los primeros 5 registros de la data

```
df.show(5)
```

id	geografia	ucll_name	ucll_id[servicio_id]	servicio_name[nro_fichas_rfo]	nro_fichas_rfo_ok	nro_sn2_sn1	dependencias	nro_sn2_sn1	nro_features_desplegadas_calidad	nro_features_desplegadas	puntaje_total_adopcion_sn2	sn2_sn1_medidos	total_vulnerabilidades_high	total_lineas_codigo
1	ARGENTINA	INGENIERIA & DATA	ALPHA	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040
2	ARGENTINA	INGENIERIA & DATA	ALPHA	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040
3	ARGENTINA	INGENIERIA & DATA	ALPHA	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040
4	ARGENTINA	INGENIERIA & DATA	ALPHA	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040
5	ARGENTINA	INGENIERIA & DATA	ALPHA	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040	4040

Task 2: Leer el esquema de la data (para verificar los tipos de datos inferidos)

```
df.printSchema()
```

```
root
 |-- mes: string (nullable = true)
 |-- geografia: string (nullable = true)
 |-- ucll_name: string (nullable = true)
 |-- ucll_id: integer (nullable = true)
 |-- servicio_name: string (nullable = true)
 |-- nro_fichas_rfo: integer (nullable = true)
 |-- nro_fichas_rfo_ok: integer (nullable = true)
 |-- nro_sn2_sn1: integer (nullable = true)
 |-- dependencias: integer (nullable = true)
 |-- nro_features_desplegadas_calidad: integer (nullable = true)
 |-- nro_features_desplegadas: integer (nullable = true)
 |-- puntaje_total_adopcion_sn2: double (nullable = true)
 |-- sn2_sn1_medidos: integer (nullable = true)
 |-- total_vulnerabilidades_high: integer (nullable = true)
 |-- total_lineas_codigo: integer (nullable = true)
```

Task 3: Leer la cantidad de registros de la data

```
df.count()
```

122

Nota. *Queries en Zeppelin: primeras filas, esquema y conteo de registros del practitioner*

➤ Cálculo de los 6 KPI's del Practitioner

```
%pyspark
from pyspark.sql import functions as F
from pyspark.sql.window import Window

# Agregar un ID de fila para mantener el orden original del CSV
df = df.withColumn("row_id", F.monotonically_increasing_id())

# Crear ventana para obtener datos del mes anterior por servicio
window_prev = Window.partitionBy("geografia", "servicio_name").orderBy("mes")

df_calc = (df
    # KPI A.a - RFO OK %
    .withColumn("rfo_ok_pct", F.when(F.col("nro_fichas_rfo") > 0,
        F.round((F.col("nro_fichas_rfo_ok")/F.col("nro_fichas_rfo"))*100, 2)))
    # KPI A.b - Dependencias %
    .withColumn("dep_pct", F.when(F.col("nro_sn2_sn1") > 0,
        F.round((F.col("nro_sn2_sn1_dependencias")/F.col("nro_sn2_sn1"))*100, 2)))
    # KPI B.a - Adopción SN2 %
    .withColumn("adopcion_sn2_pct", F.when(F.col("sn2_sn1_medidos") > 0,
        F.round((F.col("puntaje_total_adopcion_sn2")/F.col("sn2_sn1_medidos"))*100, 2)))
    # KPI C.a - Calidad de features %
    .withColumn("calidad_features_pct", F.when(F.col("nro_features_desplegadas") > 0,
        F.round((F.col("nro_features_desplegadas_calidad")/F.col("nro_features_desplegadas"))*100, 2)))
    # KPI D.a - Vulnerabilidades por mil LOC (actual)
    .withColumn("vuln_actual",
        F.when(F.col("total_lineas_codigo") > 0,
            (F.col("total_vulnerabilidades_high")/F.col("total_lineas_codigo"))*1000)
    )

# Solo calcular mes anterior si vuln_actual está entre 0.04 y 0.2
df_calc = df_calc.withColumn("necesita_mes_anterior",
    F.when(F.col("total_lineas_codigo") <= 0, False)
    .when(F.col("vuln_actual") < 0.04, False)
    .when(F.col("vuln_actual") > 0.2, False)
```

```

        .otherwise(True)
    )

# Obtener datos del mes anterior SOLO cuando se necesita
df_calc = (df_calc
    .withColumn("total_vulnerabilidades_high_prev",
        F.when(F.col("necesita_mes_anterior"),
            F.lag("total_vulnerabilidades_high").over(window_prev)))
    .withColumn("total_lineas_codigo_prev",
        F.when(F.col("necesita_mes_anterior"),
            F.lag("total_lineas_codigo").over(window_prev)))

    # Vulnerabilidades por mil LOC (anterior)
    .withColumn("vuln_anterior",
        F.when(~F.col("necesita_mes_anterior"), None)
        .when(F.col("total_lineas_codigo_prev").isNull(), F.lit(-99))
        .when(F.col("total_lineas_codigo_prev") <= 0, F.lit(-99))

    .otherwise((F.col("total_vulnerabilidades_high_prev")/F.col("total_lineas_codigo_prev"))
        *1000))

    # Evolución de vulnerabilidades
    .withColumn("evol_vuln",
        F.when(~F.col("necesita_mes_anterior"), None)
        .when(F.col("vuln_anterior") == -99, None)
        .when(F.col("vuln_anterior") == 0, None)
        .otherwise(((F.col("vuln_actual") -
            F.col("vuln_anterior"))/F.col("vuln_anterior"))*100))

    # KPI D.a - Seguridad %
    .withColumn("seguridad_pct",
        # 1. Si no hay código en mes actual, no se considera
        F.when(F.col("total_lineas_codigo") <= 0, None)
        # 2. Evaluar umbrales absolutos PRIMERO
        .when(F.col("vuln_actual") < 0.04, F.lit(100))
        .when(F.col("vuln_actual") > 0.2, F.lit(0))
        # 3. Si está entre umbrales, evaluar evolución
        .when(F.col("vuln_anterior") == -99, None)
        .when(F.col("evol_vuln").isNull(), None)
        .when(F.col("evol_vuln") == -100, F.lit(0))
        .when(F.col("evol_vuln") <= -10, F.lit(100))
        .when((F.col("evol_vuln") > -10) & (F.col("evol_vuln") < 0), F.lit(75))
        .when(F.col("evol_vuln") == 0, F.lit(50))
        .when((F.col("evol_vuln") > 0) & (F.col("evol_vuln") < 10), F.lit(25))
        .when(F.col("evol_vuln") >= 10, F.lit(0))
        .otherwise(None)
    )
)

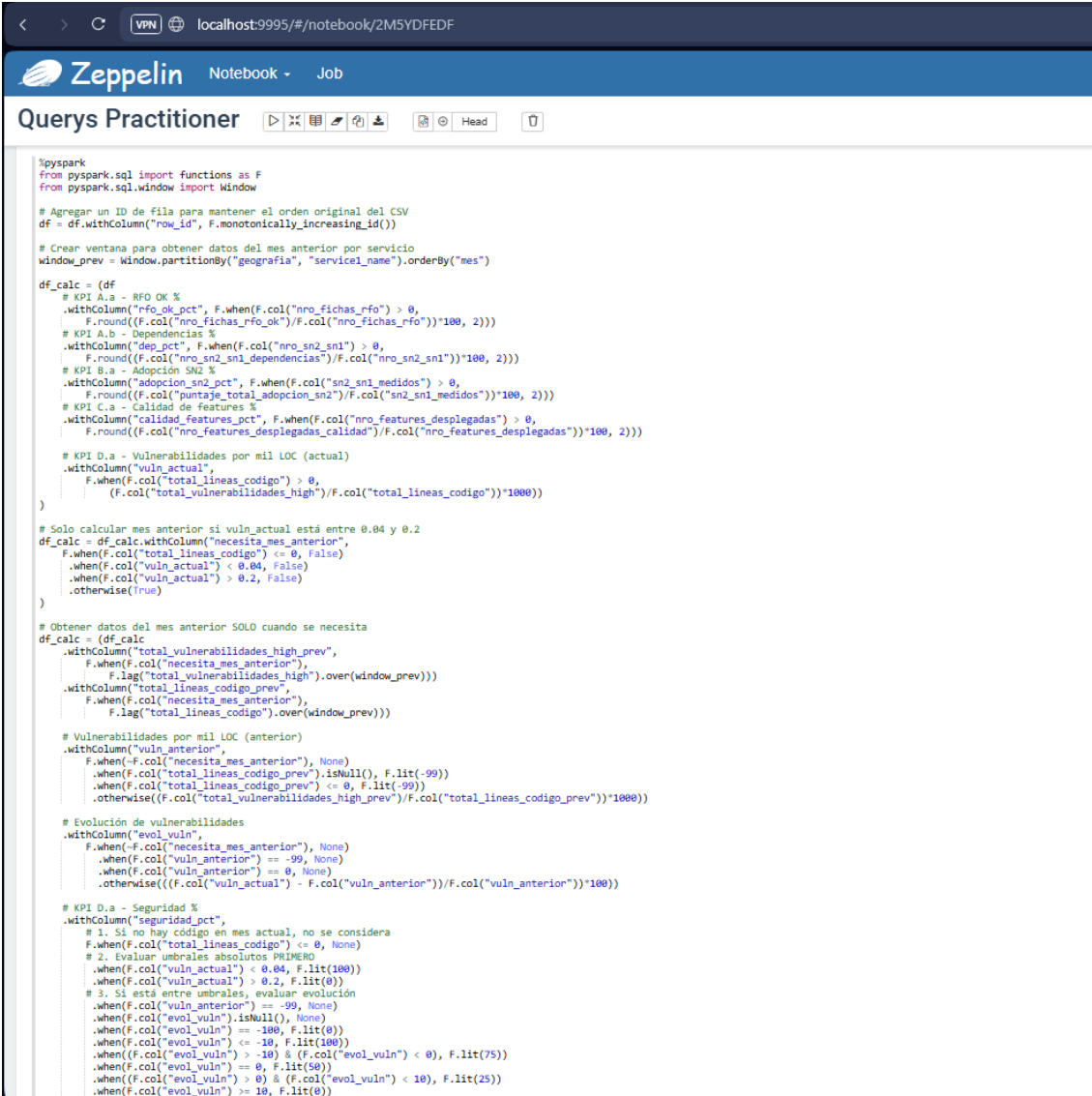
# Definición de pesos
pesos = {
    "rfo_ok_pct": 13,
    "dep_pct": 13,
    "adopcion_sn2_pct": 27,
    "calidad_features_pct": 20,
    "seguridad_pct": 27
}

# Construir columnas ponderadas
df_calc = df_calc.withColumn("suma_pesada",
    sum(F.when(F.col(k).isNotNull(), F.col(k)*v).otherwise(0) for k,v in pesos.items())
).withColumn("suma_pesos",
    sum(F.when(F.col(k).isNotNull(), F.lit(v)).otherwise(0) for k,v in pesos.items())
).withColumn("adopcion_total_pct",
    F.when(F.col("suma_pesos") > 0, F.round(F.col("suma_pesada") / F.col("suma_pesos"),
2)) .otherwise(None)
)

#Para filtrar de otras formas
df_calc.filter(F.col("nombre columna") == "nombre
especifico").orderBy("row_id").select(

```

```
# Mostrar las primeras 20 filas con todas las columnas de KPIs
df_calc.orderBy("row_id").select(
    "mes", "geografia", "service1 name",
    "rfo_ok_pct", "dep_pct", "adopcion_sn2_pct", "calidad_features_pct",
    "seguridad_pct", "adopcion_total_pct"
).show(20, truncate=False)
```



The screenshot shows a Zeppelin Notebook interface with a blue header bar containing the Zeppelin logo and the text "Notebook - Job". Below the header is a toolbar with icons for running, saving, and other actions. The main area displays a PySpark SQL query for calculating various KPIs. The query starts with imports for functions and window functions, followed by a series of window functions to calculate metrics like RFO OK %, Dependencies %, Adoption SN2 %, Features %, Vulnerabilities per mil LOC (actual), and Evolution of vulnerabilities. It also includes logic for calculating the previous month's data and a final security KPI calculation.

```
%pyspark
from pyspark.sql import functions as F
from pyspark.sql.window import Window

# Agregar un ID de fila para mantener el orden original del CSV
df = df.withColumn("row_id", F.monotonically_increasing_id())

# Crear ventana para obtener datos del mes anterior por servicio
window_prev = Window.partitionBy("geografia", "service1 name").orderBy("mes")

df_calc = (df
    # KPI A.a - RFO OK %
    .withColumn("rfo_ok_pct", F.when(F.col("nro_fichas_rfo") > 0,
        F.round((F.col("nro_fichas_rfo_ok")/F.col("nro_fichas_rfo"))*100, 2)))
    # KPI A.b - Dependencias %
    .withColumn("dep_pct", F.when(F.col("nro_sn2_sn1") > 0,
        F.round((F.col("nro_sn2_sn1_dependencias")/F.col("nro_sn2_sn1"))*100, 2)))
    # KPI B.a - Adopción SN2 %
    .withColumn("adopcion_sn2_pct", F.when(F.col("sn2_sn1_medidos") > 0,
        F.round((F.col("puntaje_total_adopcion_sn2")/F.col("sn2_sn1_medidos"))*100, 2)))
    # KPI C.a - Calidad de Features %
    .withColumn("calidad_features_pct", F.when(F.col("nro_features_desplegadas") > 0,
        F.round((F.col("nro_features_desplegadas_calidad")/F.col("nro_features_desplegadas"))*100, 2)))
    # KPI D.a - Vulnerabilidades por mil LOC (actual)
    .withColumn("vuln_actual",
        F.when(F.col("total_lineas_codigo") > 0,
            (F.col("total_vulnerabilidades_high")/F.col("total_lineas_codigo"))*1000))
)

# Solo calcular mes anterior si vuln_actual está entre 0.04 y 0.2
df_calc = df_calc.withColumn("necesita_mes_anterior",
    F.when(F.col("total_lineas_codigo") <= 0, False)
    .when(F.col("vuln_actual") < 0.04, False)
    .when(F.col("vuln_actual") > 0.2, False)
    .otherwise(True)
)

# Obtener datos del mes anterior SOLO cuando se necesita
df_calc = (df_calc
    .withColumn("total_vulnerabilidades_high_prev",
        F.when(F.col("necesita_mes_anterior"),
            F.lag("total_vulnerabilidades_high").over(window_prev)))
    .withColumn("total_lineas_codigo_prev",
        F.when(F.col("necesita_mes_anterior"),
            F.lag("total_lineas_codigo").over(window_prev)))
    # Vulnerabilidades por mil LOC (anterior)
    .withColumn("vuln_anterior",
        F.when(~F.col("necesita_mes_anterior"), None)
        .when(F.col("total_lineas_codigo_prev").isNull(), F.lit(-99))
        .when(F.col("total_lineas_codigo_prev") <= 0, F.lit(-99))
        .otherwise((F.col("total_vulnerabilidades_high_prev")/F.col("total_lineas_codigo_prev"))*1000))
    # Evolución de vulnerabilidades
    .withColumn("evol_vuln",
        F.when(~F.col("necesita_mes_anterior"), None)
        .when(F.col("vuln_anterior") == -99, None)
        .when(F.col("vuln_anterior") == 0, None)
        .otherwise((F.col("vuln_actual") - F.col("vuln_anterior"))/F.col("vuln_anterior"))*100)
    # KPI D.a - Seguridad %
    .withColumn("seguridad_pct",
        # 1. Si no hay código en mes actual, no se considera
        F.when(F.col("total_lineas_codigo") <= 0, None)
        # 2. Evaluar umbrales absolutos PRIMERO
        .when(F.col("vuln_actual") < 0.04, F.lit(100))
        .when(F.col("vuln_actual") > 0.2, F.lit(0))
        # 3. Si está entre umbrales, evaluar evolución
        .when(F.col("vuln_anterior") == -99, None)
        .when(F.col("evol_vuln").isNull(), None)
        .when(F.col("evol_vuln") == -100, F.lit(0))
        .when(F.col("evol_vuln") <= -10, F.lit(100))
        .when((F.col("evol_vuln") > -10) & (F.col("evol_vuln") < 0), F.lit(75))
        .when(F.col("evol_vuln") == 0, F.lit(50))
        .when((F.col("evol_vuln") > 0) & (F.col("evol_vuln") < 10), F.lit(25))
        .when(F.col("evol_vuln") >= 10, F.lit(0))
    )
)
```

```

    ) .otherwise(None)
  )
}

# Definición de pesos
pesos = {
  "rfo_ok_pct": 13,
  "dep_pct": 13,
  "adopcion_sn2_pct": 27,
  "calidad_features_pct": 20,
  "seguridad_pct": 27
}

# Construir columnas ponderadas
df_calc = df_calc.withColumn("suma_pesada",
  sum(F.when(F.col(k).isNotNull(), F.col(k)*v).otherwise(0) for k,v in pesos.items())
).withColumn("suma_pesos",
  sum(F.when(F.col(k).isNotNull(), F.lit(v)).otherwise(0) for k,v in pesos.items())
).withColumn("adopcion_total_pct",
  F.when(F.col("suma_pesos") > 0, F.round(F.col("suma_pesada") / F.col("suma_pesos"), 2)).otherwise(None)
)

# FILTRO: Cambia este valor para filtrar por % de adopción mínima
umbral_adopcion = 90

# Mostrar filas filtradas por umbral de adopción
df_calc.filter(F.col("adopcion_total_pct") >= umbral_adopcion).orderBy("row_id").select(
  "mes", "geografia", "service1_name",
  "rfo_ok_pct", "dep_pct", "adopcion_sn2_pct", "calidad_features_pct",
  "seguridad_pct", "adopcion_total_pct"
).show(3122, truncate=false)

```

mes	geografia	service1_name	rfo_ok_pct	dep_pct	adopcion_sn2_pct	calidad_features_pct	seguridad_pct	adopcion_total_pct
03/25/2025	ARGENTINA	ALPHA	100.0	100.0	100.0	98.77	100	99.75
03/25/2025	ARGENTINA	DAT10-AR MODELOS LOCALES	100.0	100.0	100.0	100.0	100	100.0
03/25/2025	ARGENTINA	DAT10-AR MODELOS GLOBALES	100.0	100.0	100.0	100.0	100	100.0
03/25/2025	ARGENTINA	AR21 LOANS SERVICING	100.0	100.0	100.0	100.0	100	100.0
03/25/2025	ARGENTINA	AR20 LOANS / PRESTAMOS	100.0	100.0	100.0	100.0	100	100.0
03/25/2025	ARGENTINA	AR22 ACCOUNTS & DEPOSITS / CUENTAS	100.0	100.0	66.67	100.0	100	91.0
03/25/2025	ARGENTINA	AR23 PEOPLE SERVICING	100.0	100.0	100.0	100.0	100	100.0
03/25/2025	ARGENTINA	AR26 SERVICIOS PASIVOS	100.0	100.0	100.0	100.0	null	100.0
03/25/2025	ARGENTINA	AR20 PRENDARIOS / HIPOTECARIOS	100.0	100.0	100.0	100.0	null	100.0
03/25/2025	ARGENTINA	CF06 FINANCE / FINANZAS	100.0	100.0	100.0	100.0	null	100.0
03/25/2025	ARGENTINA	AR18 AUTOMATIZACION DE PROCESO	100.0	100.0	null	100.0	100	100.0
03/25/2025	ARGENTINA	AR38 SALUD FINANCIERA	100.0	100.0	null	100.0	100	91.1
03/25/2025	ARGENTINA	AR35 BRANCHES / OFICINAS OPERATORIA	100.0	100.0	100.0	null	100	100.0
03/25/2025	ARGENTINA	AR28 PAGOS Y COBROS	100.0	83.33	90.91	100.0	100	95.38

Nota. Cálculo de los 6 KPI's del Practitioner en Zeppelin.

- Para la data cruda del Continuous Integration

➤ Mostrar las 5 primeras filas, ver el tipo de dato y conteo de filas.

```

%pyspark
csv_file =
"hdfs:///user/root/data_cruda_continuous_integration/data_cruda_continuous_integration.csv"
# Lee el archivo CSV usando el API de Spark
df = spark.read.option("header", "true").option("inferSchema", "true").csv(csv_file)

%pyspark
# Muestra las primeras filas de la data
df_subset = df.limit(8).subtract(df.limit(3))
df_subset.show()

%pyspark
# Muestra el esquema de la data (para verificar los tipos de datos inferidos)
df.printSchema()

%pyspark
# Muestra la cantidad de registros de la data
df.count()

```

Query 1
Leer el archivo CSV usando el F.read() de Spark
df = spark.read.option("header", "true").option("inferSchema", "true").csv(csv_file)

Query 2
Muestra las primeras filas de la data
df.show(10)

Query 3
Muestra el esquema de la data (para verificar los tipos de datos inferidos)
df.printSchema()

Nota. *Queries en Zeppelin: primeras filas, esquema y conteo de registros del continuous integration*

➤ Cálculo de los 13 KPI's del Continuous Integration

```
%pyspark
from pyspark.sql import functions as F
from pyspark.sql.window import Window

# Agregar un ID de fila para mantener el orden original del CSV
df = df.withColumn("row_id", F.monotonically_increasing_id())

df_calc = (df
    # A.a - % Análisis en estado «Analysis in Review» menor o igual a 7 días
    .withColumn("analysis_review_7dias_pct",
        F.when(F.col("issues_analysis_in_review") > 0,

F.round((F.col("issues_analysis_in_review_menor_7_dias")/F.col("issues_analysis_in_review")*100, 2)))

    # B.a - % Historias de usuario con Release/FixVersión asociado
    .withColumn("historias_fix_version_pct",
        F.when(F.col("nro_historias_deployed") > 0,

F.round((F.col("historias_deployed_fix_version")/F.col("nro_historias_deployed")*100, 2)))

    # C.a - % Repositorios con nomenclatura estándar
    .withColumn("repos_nomenclatura_pct",
        F.when(F.col("total_repositorios_activos") > 0,

F.round((F.col("repositorios_activos_nomenclatura_estandar")/F.col("total_repositorios_activos")*100, 2)))

    # C.b - Tiempo medio de aprobación de Pull Requests
    .withColumn("aprobacion_pr_pct",
        F.when(F.col("tiempo_medio_aprobacion_pr") < 1, F.lit(90))
        .when((F.col("tiempo_medio_aprobacion_pr") >= 1) &
(F.col("tiempo_medio_aprobacion_pr") < 1.5), F.lit(70))
        .when((F.col("tiempo_medio_aprobacion_pr") >= 1.5) &
(F.col("tiempo_medio_aprobacion_pr") < 3), F.lit(30))
```



```

        .when(F.col("tiempo_medio_aprobacion_pr") >= 3, F.lit(0))
        .otherwise(None)

# C.c - Tamaño Medio de Pull Requests
.withColumn("tamano_pr_pct",
    F.when(F.col("tamano_medio_pr") < 300, F.lit(90))
    .when((F.col("tamano_medio_pr") >= 300) & (F.col("tamano_medio_pr") <
400), F.lit(70))
    .when((F.col("tamano_medio_pr") >= 400) & (F.col("tamano_medio_pr") <
600), F.lit(30))
    .when(F.col("tamano_medio_pr") >= 600, F.lit(0))
    .otherwise(None))

# C.d - Repositorios gobernados en el análisis estático de Seguridad
.withColumn("repos_chimera_pct",
    F.when(F.col("total_repositorios_activos") > 0,

F.round((F.col("repositorios_activos_gobernados_chimera")/F.col("total_repositorios_acti
vos"))*100, 2)))

# D.a - Tiempo medio de integración
.withColumn("tiempo_integracion_pct",
    F.when(F.col("tiempo_medio_integracion") < 3, F.lit(90))
    .when((F.col("tiempo_medio_integracion") >= 3) &
(F.col("tiempo_medio_integracion") < 4), F.lit(70))
    .when((F.col("tiempo_medio_integracion") >= 4) &
(F.col("tiempo_medio_integracion") < 6), F.lit(30))
    .when(F.col("tiempo_medio_integracion") >= 6, F.lit(0))
    .otherwise(None))

# D.b - Tiempo medio construcciones
.withColumn("tiempo_construcciones_pct",
    F.when(F.col("tiempo_medio_construcciones") < 20, F.lit(90))
    .when((F.col("tiempo_medio_construcciones") >= 20) &
(F.col("tiempo_medio_construcciones") < 35), F.lit(70))
    .when((F.col("tiempo_medio_construcciones") >= 35) &
(F.col("tiempo_medio_construcciones") < 45), F.lit(30))
    .when(F.col("tiempo_medio_construcciones") >= 45, F.lit(0))
    .otherwise(None))

# D.c - % Construcciones correctas
.withColumn("construcciones_correctas_pct",
    F.when(F.col("ejecuciones_pipeline") > 0,

F.round((F.col("ejecuciones_pipeline_success")/F.col("ejecuciones_pipeline"))*100, 2)))

# D.d - Tiempo medio en arreglar construcciones
.withColumn("tiempo_arreglar_construcciones_pct",
    F.when(F.col("tiempo_medio_arreglar_construcciones") < 60, F.lit(90))
    .when((F.col("tiempo_medio_arreglar_construcciones") >= 60) &
(F.col("tiempo_medio_arreglar_construcciones") < 120), F.lit(70))
    .when((F.col("tiempo_medio_arreglar_construcciones") >= 120) &
(F.col("tiempo_medio_arreglar_construcciones") < 180), F.lit(30))
    .when(F.col("tiempo_medio_arreglar_construcciones") >= 180, F.lit(0))
    .otherwise(None))

# E.a - Calidad del código
.withColumn("calidad_codigo_pct",
    F.when(F.col("repositorios_activos_sonarqube") > 0,

F.round((F.col("repositorios_activos_sonarqube_ok")/F.col("repositorios_activos_sonarqub
e"))*100, 2)))

# E.b - % Historias de Usuario, Dependencias y Bugs con pruebas de aceptación (XRay)
.withColumn("items_pruebas_xray_pct",
    F.when(F.col("items_desplegados") > 0,

F.round((F.col("items_desplegados_pruebas_xray")/F.col("items_desplegados"))*100, 2)))
)

# Definición de pesos (suman 100%)
pesos_ci = {

```

```

"analisis_review_7dias_pct": 5,
"historias_fix_version_pct": 6,
"repos_nomenclatura_pct": 6,
"aprobacion_pr_pct": 10,
"tamano_pr_pct": 5,
"repos_chimera_pct": 7,
"tiempo_integracion_pct": 16,
"tiempo_construcciones_pct": 9,
"construcciones_correctas_pct": 13,
"tiempo_arreglar_construcciones_pct": 8,
"calidad_codigo_pct": 10,
"items_pruebas_xray_pct": 5
}

# Construir columnas ponderadas para Continuous Integration
df_calc = df_calc.withColumn("suma_pesada_ci",
    sum(F.when(F.col(k).isNotNull(), F.col(k)*v).otherwise(0) for k,v in
pesos_ci.items())
).withColumn("suma_pesos_ci",
    sum(F.when(F.col(k).isNotNull(), F.lit(v)).otherwise(0) for k,v in pesos_ci.items())
).withColumn("adopcion_ci_pct",
    F.when(F.col("suma_pesos_ci") > 0, F.round(F.col("suma_pesada_ci") /
F.col("suma_pesos_ci"), 2)).otherwise(None)
)

# FILTRO: Cambia este valor para filtrar por % de adopción mínima
umbral_adopcion = 90

# Mostrar filas filtradas por umbral de adopción
df_calc.filter(F.col("adopcion_ci_pct") >= umbral_adopcion).orderBy("row_id").select(
    "mes","geografia","sn1",
    "analisis_review_7dias_pct","historias_fix_version_pct","repos_nomenclatura_pct",
    "aprobacion_pr_pct","tamano_pr_pct","repos_chimera_pct",
    "tiempo_integracion_pct","tiempo_construcciones_pct","construcciones_correctas_pct",
    "tiempo_arreglar_construcciones_pct","calidad_codigo_pct","items_pruebas_xray_pct",
    "adopcion_ci_pct"
).show(1326, truncate=False)

```

[illegible]

3. REFERENCIAS BIBLIOGRÁFICAS

Apache Hadoop Project. (2025). <https://hadoop.apache.org/>
 Apache Software Foundation. (2025). Apache Spark documentation. Recuperado de:
<https://spark.apache.org/docs/4.0.1/index.html>
 Apache Spark. (2025). <https://spark.apache.org/>
 Hortonworks / Cloudera. (2016). Hortonworks Data Platform: Apache Ambari User
 Guide. [https://docs-archive.cloudera.com/HDPDocuments/Ambari-
 2.4.2.0/bk_ambari-user-guide/bk_ambari-user-guide.pdf](https://docs-archive.cloudera.com/HDPDocuments/Ambari-2.4.2.0/bk_ambari-user-guide/bk_ambari-user-guide.pdf)
 Sharda, R., & Delen, D. & Turban, E. (2020). Analytics, Data Science, & Artificial
 Intelligence Systems for Decision Support. 11th ed. Pearson.
[https://api.pageplace.de/preview/DT0400.9781292341606_A39573369/preview-
 9781292341606_A39573369.pdf](https://api.pageplace.de/preview/DT0400.9781292341606_A39573369/preview-9781292341606_A39573369.pdf)