**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS**

Logotipo

El contenido generado por IA puede ser incorrecto.

**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 Enginering.
* 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** |
| Enginering | Service Owner | ¿Qué servicios N1 tienen mayor porcentaje de adopción del nivel Practitioner y cuáles requieren intervención inmediata? | Alta | Nucleus, Continuum, JIRA, Chimera, Bitbucket |
| Enginering | 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 |
| Enginering | 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 | ¿Cuál es 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, GIAM |
| 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 | ¿Cuál es la evolución mensual de vulnerabilidades por cada mil líneas de código y qué servicios representan mayor riesgo de seguridad? | Alta | Chimera, Bitbucket |
| Enginering | Gerente de Área | ¿Qué unidades organizacionales (UOL) tienen menor adopción de prácticas ágiles y requieren planes de mejora prioritarios? | Alta | Todos los sistemas integrados |
| Enginering | Service Owner | ¿Cuál es el porcentaje de construcciones correctas y tiempo medio de arreglo de fallos por servicio? | Media | Jenkins |
| Planeación Estratégica | Directorio/ Gerencia General | ¿Cuál es 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 |
| Enginering | 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

**Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.**

**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

Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

**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 | * 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 | \* Si hay una división entre cero en la fórmula, no se considera el indicador. |
| 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 | \* Si hay una división entre cero en la fórmula, no se considera el indicador. |
| 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 | \* 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) Feactures 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 | * 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 | \* 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 | * 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 | * Vulnerabilidad (cada mil líneas de código): * Evolución de Vulnerabilidades:   \* 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%. |
| 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) | * 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 | \* 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. |
| 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 | \* 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/FixVersión asociado**

|  |  |
| --- | --- |
| **CAMPO** | **DESCRIPCIÓN / EJEMPLO** |
| Nombre del KPI | % Historias de usuario con Release/FixVersión asociado |
| Objetivo estratégico asociado | Fomentar el uso de *releases* 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 *«Deployed»* durante el mes de medición que tienen el campo *«Fix Version»* informado. |
| Fórmula | \* 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 | 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 | \* Si hay una división entre cero en la fórmula, no se considera el indicador. |
| 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 % |
| Umbrales (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 |  |
| Unidad de Medida | % |
| Frecuencia de Medición | Mensual |
| Fuente de Datos | Chimera, GIAM, 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 | 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 | * Identificar cuellos de botella en revisiones, pruebas, pipelines o dependencias que ralentizan la integración. * 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 % |
| Umbrales (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 | * 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 % |
| Umbrales (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 | * 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 | \* Si hay una división entre cero en la fórmula, no se considera el indicador. |
| 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 | * 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 | \* Si hay una división entre cero en la fórmula, no se considera el indicador. |
| 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 | * 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 | \* 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) |

**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 | \* 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. |
| 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

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**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

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

**Fuente:** *Elaboración propia.*

**Figura 5.** Modelo Conceptual Estrella de la data Continuous Integration

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

**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.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Nota.** *Interfaz SandBox*

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

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**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

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

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

Texto

El contenido generado por IA puede ser incorrecto.

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

d) Ver la versión del Spark en Web Shell Client

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**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.

Texto

El contenido generado por IA puede ser incorrecto.

**Nota.** *Interfaz Web Shell Client creando las carpetas*

Una captura de pantalla de una red social

El contenido generado por IA puede ser incorrecto.

**Nota.** *Interfaz Files View con las carpetas creadas*

Una captura de pantalla de una red social

El contenido generado por IA puede ser incorrecto.

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

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**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,

uol1\_name STRING,

uol2\_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*

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**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");

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

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

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**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()

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**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", "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)

)

#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",

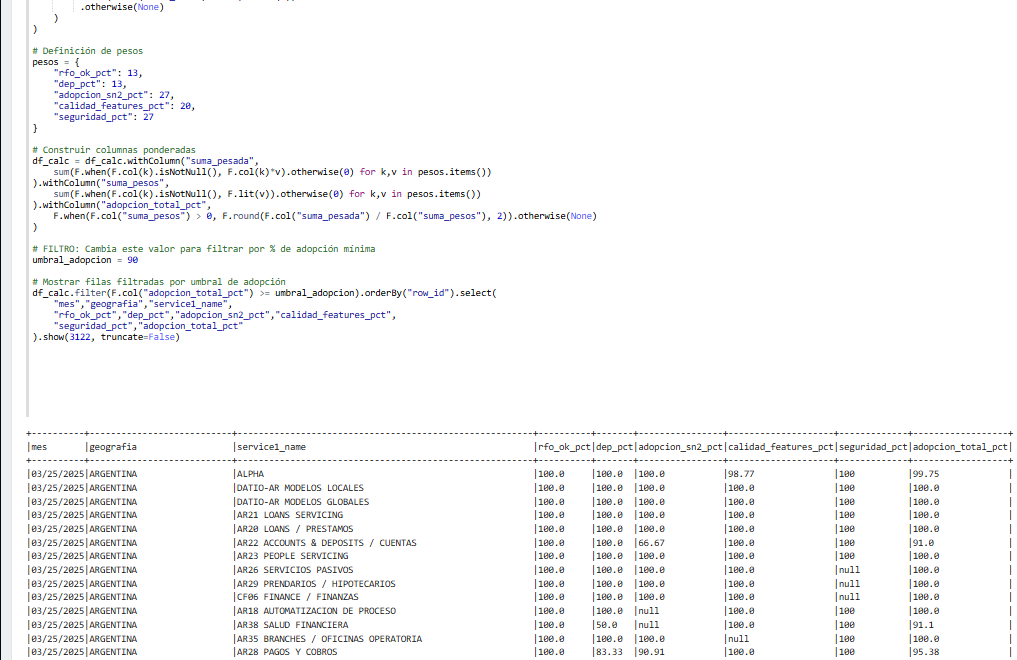
"rfo\_ok\_pct","dep\_pct","adopcion\_sn2\_pct","calidad\_features\_pct",

"seguridad\_pct","adopcion\_total\_pct"

).show(20, truncate=False)

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.



**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()

Tabla

El contenido generado por IA puede ser incorrecto.

**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("analisis\_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\_activos"))\*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\_sonarqube"))\*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)

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

**Nota.** *Cálculo de los 13 KPI’s y adopción total en Continuous Integration.*

**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>