



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

Experience Report: ReportsHub

Tesis de Licenciatura en Ciencias de la Computación

Pablo S. Casullo

Director: Diego Garbervetsky

Buenos Aires, 2025

EXPERIENCE REPORT: REPORTSHUB

En las grandes corporaciones multinacionales coexisten múltiples tecnologías y prácticas de Business Intelligence (BI) que abarcan todo el espectro del stack tecnológico, desde el Data Warehouse (DW), pasando por los procesos de Extract, Transform and Load (ETL) [4] hasta herramientas de visualización y análisis. A pesar de los intentos por establecer estándares tecnológicos y de Gobierno [5], la descentralización de iniciativas entre áreas funcionales y niveles geográficos genera un ecosistema complejo, heterogéneo y difícil de armonizar. Este escenario plantea importantes desafíos tanto en la integración de soluciones como en la medición de su impacto en el negocio, donde la relación entre adopción e valor resulta poco evidente[1]. En este contexto, definir Key Performance Indicators (KPIs) de adopción se vuelve crítico para gestionar de manera eficiente el portafolio de soluciones de BI, aunque su implementación requiere superar barreras metodológicas y organizacionales. Asimismo, la User Experience (UX) se ve afectada por la falta de consistencia en el manejo de los accesos y servicios, lo que refuerza la necesidad de establecer enfoques más integrados y estandarizados en la gestión de estas herramientas.

Estas corporaciones, suelen tener prácticas de definiciones de estándares tecnológicos, que definen herramientas o tecnologías oficiales a ciertos productos y a proveedores, con el fin de simplificar y lograr sinergías tecnológicas entre las herramientas seleccionadas y reducir las combinaciones, según distintos casos de uso[6].

Palabras claves: *BI, DW, Stack tecnológico, UX, Portfolio management, Automatización, Decentralized Access Management, Global process, Gobierno.*

AGRADECIMIENTOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sapien ipsum, aliquet eget convallis at, adipiscing non odio. Donec porttitor tincidunt cursus. In tellus dui, varius sed scelerisque faucibus, sagittis non magna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Mauris et luctus justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris sit amet purus massa, sed sodales justo. Mauris id mi sed orci porttitor dictum. Donec vitae mi non leo consectetur tempus vel et sapien. Curabitur enim quam, sollicitudin id iaculis id, congue euismod diam. Sed in eros nec urna lacinia porttitor ut vitae nulla. Ut mattis, erat et laoreet feugiat, lacus urna hendrerit nisi, at tincidunt dui justo at felis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut iaculis euismod magna et consequat. Mauris eu augue in ipsum elementum dictum. Sed accumsan, velit vel vehicula dignissim, nibh tellus consequat metus, vel fringilla neque dolor in dolor. Aliquam ac justo ut lectus iaculis pharetra vitae sed turpis. Aliquam pulvinar lorem vel ipsum auctor et hendrerit nisl molestie. Donec id felis nec ante placerat vehicula. Sed lacus risus, aliquet vel facilisis eu, placerat vitae augue.

A mi persona favorita.

Índice general

1.. Motivacion	1
1.1. Contexto organizacional	1
1.2. Situación inicial y limitaciones detectadas	1
1.3. Problemas principales para usuarios y equipos de datos	1
1.3.1. Gestión de accesos	1
1.3.2. Consumo de contenidos	2
1.3.3. Manejo del portafolio de soluciones de inteligencia de negocios	2
1.4. Aporte de la Tesis	3
2.. Definiciones Preliminares	5
2.1. Definiciones de industria y corporaciones multinacionales	5
2.2. Definiciones de inteligencia de negocios	5
2.3. Definiciones de diseño y arquitectura de software	5
2.4. Definiciones de metodologías	6
2.5. Definiciones propias del proyecto	6
3.. Propuesta	9
3.1. Visión y objetivos del proyecto	9
3.2. Principios de diseño de ReportHub	9
3.2.1. Web application	9
3.2.2. lean UX minimalista y defensiva	10
3.2.3. Tener una Service Oriented Architecture (SOA)	10
3.2.4. Descentrado y Escalable	10
3.2.5. Auditable	10
3.3. Casos de uso principales	10
3.3.1. Creación de estructuras geográficas para organizar los contenidos (Admin Global).	11
3.3.2. Creación de áreas de información para organizar los contenidos(Admin Global/Local).	11
3.3.3. Creación y administración de Usuarios.	11
3.3.4. Publicación de contenidos (Creador/Admin).	12
3.3.5. Descubrimiento y navegación en el catálogo (Consumidor).	13
3.3.6. Circuito de Aprobaciones (Creador/Admin).	14
3.3.7. Navegación, acceso al portal y a reportes y monitoreo de estadísticas.	14
3.4. Alcance inicial y entregables previstos	15
3.5. Definición de KPIs y métricas de éxito	15
4.. Arquitectura	17
4.1. Principios de diseño	17
4.2. Arquitectura	19
4.3. Componentes principales	19
4.4. Diseño de seguridad y Row Level Security (RLS)	21
4.5. Flujos de datos e integraciones con sistemas existentes	22

4.6.	Consideraciones de escalabilidad y resiliencia	22
5..	desarrollo y despliegue	23
5.1.	Enfoque metodológico (Waterfall/Agile)	23
5.2.	Planificación y backlog inicial	24
5.3.	Definición de historias de usuario clave	24
5.4.	Iteraciones y sprints principales	24
5.5.	Validaciones con usuarios de negocio	24
5.6.	Gestión de calidad	24
5.6.1.	Pruebas funcionales	24
5.6.2.	Pruebas de seguridad	24
5.6.3.	Pruebas de performance	24
5.6.4.	Pruebas de integración	24
5.7.	Hitos alcanzados y ajustes sobre la marcha	24
5.8.	Estrategia de despliegue y fases de rollout.	24
5.9.	Piloto inicial y resultados	24
5.10.	Escalamiento a más áreas y usuarios	24
5.11.	Plan de comunicación y capacitación	24
5.12.	Materiales de soporte y guías de usuario	24
5.13.	Gestión del cambio organizacional	24
5.14.	Mecanismos de soporte post-lanzamiento	24
6..	evaluacion	25
6.1.	Metodología de evaluación	25
6.2.	Resultados cuantitativos	25
6.3.	Resultados cualitativos	25
6.4.	Comparación con KPIs iniciales	25
6.5.	Impacto en seguridad y cumplimiento	25
6.6.	Conclusiones de la evaluación	25
7..	Conclusiones	27
7.1.	Resumen de los logros principales	27
7.2.	Relevancia del proyecto para la organización	27
7.3.	Impacto estratégico a largo plazo	27
7.4.	Próximos pasos sugeridos (roadmap evolutivo)	27
8..	Lecciones Aprendidas	29
8.1.	Aspectos que funcionaron bien	29
8.2.	Dificultades y cómo se superaron	29
8.3.	Factores clave de éxito	29
8.4.	Aspectos a mejorar	29
8.5.	Aprendizajes organizacionales	29
8.6.	Recomendaciones para futuros proyectos	29

1. MOTIVACION

1.1. Contexto organizacional

En las grandes corporaciones multinacionales, existen múltiples tecnologías para resolver distintos desafíos de análisis e inteligencia de negocios, a lo largo de todo el stack tecnológico que va desde los sistemas ETL [4], distintos repositorios de almacenamiento para datos estructurados y no estructurados, y herramientas de exploración y explotación de datos que se asocian a la capa de presentación. En esta última coexisten planillas de cálculo (xls) y software específico de visualización para crear reportes y tableros de información, tales como Microstrategy, Cognos, PowerBI, Tableau o Qlik, entre otros.

Estas corporaciones suelen definir estándares tecnológicos que determinan herramientas o proveedores oficiales, con el fin de simplificar y lograr sinergias entre las tecnologías seleccionadas y reducir las combinaciones según distintos casos de uso [6].

Adicionalmente, es común que las iniciativas de inteligencia de negocios estén descentralizadas y distribuidas tanto en áreas funcionales (marketing, ventas, compras, finanzas), como en distintos niveles geográficos, que abarcan desde equipos corporativos hasta instancias globales, regionales y locales.

1.2. Situación inicial y limitaciones detectadas

Este contexto de distribución de recursos en áreas y geografías, combinado con la multiplicidad de herramientas “aprobadas” por dichas corporaciones, genera una combinación de componentes para la creación de distintas soluciones que, aun siguiendo las mejores prácticas de gobernanza tecnológica y de arquitectura [5], devienen en un ecosistema complejo, desarmonizado y lleno de particularidades.

Poder establecer el valor real e impacto que estas soluciones tienen en el negocio es un desafío, ya que en muchos casos la relación entre adopción e impacto es indirecta y no trivial [1]. A su vez, la experiencia de los distintos usuarios varía significativamente: solicitar acceso o encontrar los enlaces de cada herramienta resulta inconsistente, incoherente y dependiente de qué equipo haya desarrollado la solución.

1.3. Problemas principales para usuarios y equipos de datos

1.3.1. Gestión de accesos

La gestión de accesos junto con la aplicación de políticas de seguridad una vez otorgados los mismos, variaban absolutamente y quedaban definidas por el criterio de cada equipo que desarrollaba los contenidos, sin haber coherencia alguna. Algunos ejemplos de variantes incluyen:

Enviar un email a quien dio las capacitaciones (en caso que las haya habido). Muchas veces la persona encargada de la instrucción o entrenamiento de usuarios era también la encargada de controlar el acceso y otorgarlo. En otros casos, actuaba como intermediario para poder llegar a quien era responsable de otorgar los accesos adecuados. Solicitar acceso mediante herramientas de tickets a equipos encargados de la operación de dichos tableros/reportes. En equipos de proyecto de tamaños medio o grandes (con un número

de integrantes excediendo la decena), es común que haya gente dedicada exclusivamente a tareas operativas, cuyo objetivo es por un lado garantizar la continuidad y máxima disponibilidad de las soluciones como así también en ocasiones se encargan de ejecutar las actividades que habilitan acceso y perfiles de seguridad adecuados a los usuarios. Solicitar acceso a alguien conocido si puede hacer de nexo para dar con el contacto indicado.

1.3.2. Consumo de contenidos

Para poder consumir o utilizar reportes es fundamental saber de su existencia, conocer su ubicación (normalmente son enlaces dentro de redes internas) y tener acceso a los mismos. Algunos ejemplos acerca de cómo acceder, pueden incluir:

1. Usuarios reciben enlaces de acceso por emails o en documentos de entrenamientos y luego (con suerte) los almacenan como atajos en sus navegadores (con los problemas de tener hardcoded links que luego con el tiempo pueden variar, apuntar a versiones obsoletas de dichos reportes o incluso quedar deprecados conforme algunos reportes son decomisionados).
2. Páginas de intranet donde se publican los puntos de acceso (muchas veces implementadas como portales de acceso, provistos por las mismas herramientas de visualización).
3. Usuarios reciben reportes como archivos adjuntos en correos electrónicos.

1.3.3. Manejo del portafolio de soluciones de inteligencia de negocios

Para poder hacer un manejo efectivo de un portafolio de soluciones y de las tecnologías que se utilizan, es necesario poder entender:

- ¿Qué soluciones hay disponibles y en qué tecnologías están desarrolladas (inventario)?
- ¿Qué áreas de negocio cubren y quienes son responsables?
- ¿Qué nivel de adopción tienen?
- ¿Qué usuarios tienen acceso y no deberían?
- ¿Qué usuarios necesitan acceso y no tienen?
- ¿Qué usuarios tienen acceso y no lo usan?
- ¿Qué usuarios tienen altos nivel de adopción?
- ¿Qué volumen de pedidos de acceso manejan, según las audiencias objetivo?
- ¿Hay duplicidad de contenidos, hechos por áreas afines pero sin colaborar?
- ¿Hay patrones de uso de reportes que tengan correlación con el desempeño de áreas de negocio?

La respuesta a cada una y en particular a todas estas preguntas, en el entorno descripto, es de un esfuerzo que no permite tener información en tiempo y forma ni de modo adecuado, repetible y consistente de manera constante. Poder responder cada pregunta en un contexto tan heterogéneo, implica un nivel de trabajo manual, armonización y alineación de criterios y definiciones que simplemente lo vuelven imposible, en la escala de estas organizaciones.

En resumen, los impactos negativos descriptos anteriormente pueden sintetizarse en la siguiente lista:

- Falta de un repositorio central y común donde buscar/encontrar y solicitar acceso a los contenidos necesarios genera dificultad para poder dar con los contenidos.
- Procesos inconsistentes y altamente variables.
- Falta de transparencia en cuanto a puntos de contactos de reportes.
- Métodos de aprobación de múltiples pasos y manuales, que dependen de horarios laborales, feriados distribuidos en varias zonas horarios y múltiples geografías.
- Separación entre nivel de aprobación (hecha por la persona responsable) de un acceso y el nivel de ejecución de dicho acceso (hecha por operadores, una vez que la persona responsable ha definido qué acceso corresponde).
- Falta de información necesaria para el aprobador, acerca de rol, país, función y otros elementos que ayudan a determinar si un acceso debe o no ser otorgado a un usuario.
- La ausencia de un repositorio centralizado agrava los problemas de consistencia y coherencia. Esto se traduce en dificultades para asegurar estándares de seguridad homogéneos, en la duplicación de esfuerzos.
- El estado fragmentado limita la capacidad de contar con indicadores confiables y oportunos para la toma de decisiones de portafolio.
- La falta de métricas unificadas de adopción impide evaluar el impacto de las soluciones desarrolladas, dificultando la gestión adecuada del portafolio de inteligencia de negocios.

1.4. Aporte de la Tesis

Esta *tesis*, en formato de Experience Report, tiene como objetivo, profundizar en el proceso de desarrollo de una solución que atiende a los desafíos mencionados, detallando desde su concepción en el contexto organizacional, hasta su despliegue y medición del impacto, dentro de una organización de estas características.

Los contenidos a desarrollar incluyen:

Definiciones Preliminares: Conjunto de definiciones básicas que nos introducen en el dominio del problema.

Propuesta: Descripción detallada conceptual de la solución, en función de sus requerimientos funcionales y no funcionales. [10] [8]

Arquitectura: Descripción de los componentes y sus relaciones como también algunos patrones de diseño aplicados. [9] [2] [3]

Desarrollo y Despliegue: Explicación de la metodología aplicada y entregas iterativas.

Evaluación: Monitoreo y evaluación de las métricas clave y factores de éxito.

Lecciones aprendidas: Hallazgos y aprendizajes de la experiencia en el proyecto.

Conclusiones: Síntesis final del trabajo, con implicancias finales.

2. DEFINICIONES PRELIMINARES

Las definiciones preliminares son aquellas que nos permiten profundizar en los elementos básicos del dominio de nuestro contexto. A continuación se presentan las que aplican a conceptos, roles y elementos clave utilizados a lo largo de este trabajo.

2.1. Definiciones de industria y corporaciones multinacionales

KPIs: Key Performance Indicators (Indicadores clave de desempeño).

2.2. Definiciones de inteligencia de negocios

Inteligencia de negocios: (Business intelligence)

Data Warehouse:

2.3. Definiciones de diseño y arquitectura de software

Stack tecnológico:

Web App:

RLS:

LDAP:

API:

SOA:

Cohesión:

Acoplamiento:

Web Server:

Load Balancer:

Web Application Server:

Back end:

Client Server:

Caché:

Tolerancia a fallas:

NFS:

Database Server:

MVC: Model View Controller [7]. Patrón de diseño mediante el cual, se separan en tres capas bien definidas, para mayor flexibilidad, mantenibilidad y escalabilidad.

2.4. Definiciones de metodologías

Waterfall/Cascada: Descripción de la Metodología waterfall.

Agile/Agil: Descripción de la Metodología ágil.

DevOps: Metodología DevOps.

Unit tests: Pruebas unitarias.

Integration tests: Pruebas de integración.

Dev: Desarrollo. Entorno para desarrollo y pruebas de unidad y de integración.

UAT: User Acceptance Test. Entorno de pruebas de usuario y validaciones finales antes de hacer pasajes a producción.

Pr: Production/Producción. Entorno de operaciones productivo.

2.5. Definiciones propias del proyecto

ReportHub: Plataforma propuesta para consolidar y armonizar la publicación de reportes y tableros, gestión de accesos y monitoreo de métricas de uso de BI.

Administradores: Usuarios con privilegios para gestionar contenidos, accesos y usuarios dentro de la plataforma. Se distinguen tres tipos:

- **Admin Global**: Gestiona unidades geográficas globales y áreas de información a nivel global.
- **Admin Local**: Gestiona áreas de información dentro de su región geográfica asignada.

Creadores de Contenidos: Usuarios responsables de generar y publicar contenidos dentro de la plataforma.

Consumidores de Contenidos: Usuarios que acceden y utilizan los contenidos publicados en el portal, pudiendo también marcar favoritos y realizar solicitudes de acceso.

Aprobadores: Usuarios que evalúan y aprueban solicitudes de acceso a contenidos según la configuración de seguridad definida.

Estructura de contenidos: Jerarquía de organización de contenidos en dos niveles:

1. **Nivel 1**: Unidades geográficas (Global, Europa, América, Asia, África y países correspondientes).
2. **Nivel 2**: Áreas de información (temáticas específicas, únicas dentro de cada región).

Metadatos de Contenido: Información asociada a cada contenido publicado, incluyendo:

- Título
- Descripción

- Imagen miniatura
- Tipo de contenido (archivo o URL)
- Segmentos de usuarios y objetivos de frecuencia de uso
- Aprobadores
- Configuración de provisión de acceso

3. PROPUESTA

3.1. Visión y objetivos del proyecto

Por todo lo anteriormente descrito, es que surge este proyecto, impulsando un espacio de consolidación y armonización de publicación, gestión de accesos y monitoreo de métricas de uso para la adecuada gestión del portafolio de BI. Asimismo, unificar el acceso y ofrecer una experiencia homogénea y consistente, independiente de la tecnología de implementación, se presenta como una oportunidad para mejorar tanto la eficiencia de los equipos de datos como la satisfacción de los usuarios finales.

De la complejidad de estas problemáticas surgió la necesidad de lograr una solución que permita:

Concentrar en un solo lugar la oferta de reportes/tableros ofrecida por los distintos equipos. Permitir a los equipos que publican estos elementos, realizar una gestión de sus usuarios, incluyendo la asignación de roles si aplican restricciones de visibilidad de datos. Establecer objetivos de adopción y medirlos de modo consistente, de manera totalmente independiente a la tecnología de implementación que se haya utilizado. Darle a los potenciales usuarios una experiencia homogénea y consistente, de modo agnóstico a las tecnologías utilizadas mediante técnicas de ingeniería de software. A los usuarios, almacenar atajos o accesos directos “resilientes” que resistan cambios de URLs de reportes y a la vez conserven atributos de seguridad para que compartiendo los enlaces la seguridad se mantenga.

3.2. Principios de diseño de ReportHub

El diseño de la solución se basó en un conjunto de principios arquitectónicos que aseguraron no solo la cobertura de las necesidades funcionales del momento, sino también la capacidad de evolucionar en el tiempo. Estos principios tuvieron como propósito garantizar que la plataforma fuera escalable, permitiendo crecer en volumen de usuarios, contenidos y procesos sin comprometer el rendimiento. A su vez, aseguraron un mantenimiento eficiente, reduciendo la complejidad técnica y posibilitando la incorporación de nuevas funcionalidades de forma ágil y con bajo costo operativo. Otro eje fundamental fue la facilidad de uso, que permitió que cada rol interactuara con la solución de manera simple, intuitiva y orientada a sus tareas principales, evitando barreras de adopción. Asimismo, se estableció un marco que permitió cumplir con las normas internas de auditoría, control de calidad y gobierno de la información, garantizando la trazabilidad y responsabilidad en cada acción ejecutada dentro del sistema. Finalmente, los principios incorporaron las mejores prácticas en seguridad y estándares internacionales, asegurando la protección de datos, la correcta gestión de accesos y la interoperabilidad con diferentes tecnologías, lo que fortaleció la resiliencia y confiabilidad de la solución en contextos de negocio dinámicos y regulados.

La solución debe cumplir con los siguientes principios de diseño:

3.2.1. Web application

Será una aplicación web, que contará con una elementos de presentación el el browser, una capa de lógica de negocio en un servidor y se apoyará en una base de datos relacional

para almacenar la información y meta información necesaria para la configuración, uso y auditoría de actividades.

3.2.2. lean UX minimalista y defensiva

Experiencia de usuario simple e intuitiva para cada uno de los roles en que los usuarios operen la solución, ya que un mismo usuario, puede tener contenidos para publicar, ser administrador del sistema y eventualmente consumidor de contenidos publicados por otros usuarios:

- *Administradores, roles:* “Admin Global / Admin Local”.
- *Creadores de Contenidos, rol:* “Creador”.
- *Consumidores de contenidos, rol:* “Consumidor”.

La interfaz, debe evitar que el usuario cargue información inválida mediante reglas de negocio intuitivas y ya incorporadas a la navegación y cuando esto nos sea posible, validará la información que el usuario ingrese y proveerá feedback inmediato en el momento de las interacciones para poder corregir problemas.

3.2.3. Tener una SOA

Ser agnóstico de las tecnologías en las que se han producido los contenidos que se publicarán minimizando el acoplamiento técnico y a la vez funcionando con una alta cohesión. Este punto permite garantizar la “interoperabilidad con diferentes tecnologías y a la vez el soporte de procesos comunes, sin estar condicionados por las tecnologías de implementación.

3.2.4. Descentrado y Escalable

Permitir la descentralización de la gestión de contenidos, accesos y configuraciones de modo de poder asignar en distintos niveles y equipos organizacionales las facultades para el auto servicio de sus contenidos. Esto evita los cuellos de botella de estructuras centralizadas y fomenta que se tomen las decisiones adecuadas en cada lugar adecuado de la organización, siguiendo procesos consistentes garantizados por el sistema.

3.2.5. Auditable

Cada acción debe ser auditable, para poder cumplir con normas internas de auditorías de calidad de procesos y responsabilidad, en particular a la hora de administrar usuarios y accesos.

3.3. Casos de uso principales

Los principales casos de uso, tienen como objetivo detallar en un nivel conceptual, las funcionalidades básicas y escenarios que iba a soportar el sistema.

3.3.1. Creación de estructuras geográficas para organizar los contenidos (Admin Global).

Los contenidos del sistema debían estar organizados en una jerarquía de dos niveles. El primer nivel, compuesto de unidades geográficas, tenía como objetivo agrupar por geografía y a su vez, poder otorgar niveles de administración descentralizados a distintos Admins de geografías para administrar áreas de información. Se utilizan como valores válidos, los correspondientes a las distintas unidades geográficas: -Global -Europa -America -Asia -Africa y luego países Todos estarán agrupados en un mismo nivel.

3.3.2. Creación de áreas de información para organizar los contenidos (Admin Global/Local).

El segundo nivel de organización son áreas de información y permiten una agrupación lógica de los contenidos en función de las temáticas que cubren, como por ejemplo finanzas, marketing, ventas, etc. No existe un listado predefinido de qué áreas de información existen a nivel local y pueden ser creadas libremente por los administradores a nivel geográfico. La única restricción: los nombres de las áreas en una región geográfica deben ser únicos.

Por ejemplo:

- Global
 - Marketing
 - Cumplimiento
 - Ventas
 - Finanzas
 - Recursos Humanos, etc.
- America
 - Marketing
 - Legales, etc.
- Argentina
 - Fuerza de Ventas
 - Finanzas
 - Oportunidades, etc.

3.3.3. Creación y administración de Usuarios.

La “creación” y administración de usuarios, consiste en poder registrar usuarios que ya son parte del directorio corporativo como usuarios de este sistema y asignarles roles de Admins globales, locales, de áreas de información y/o eventualmente como consumidores. Del directorio corporativo se importan los siguientes datos: handle de usuario de la compañía, nombre completo, dirección de correo electrónico.

Es importante aclarar que un Admin tiene la capacidad de crear secciones, contenidos y manejar usuarios, pero no necesariamente tiene acceso a los mismos contenidos que

Función/rol	Admin global	Admin local	Creador	Aprobador
Crear áreas globales	✓			
Crear áreas locales	✓	✓		
Crear contenidos	✓	✓	✓	
Administrar usuarios	✓	✓	✓	✓
Asignar Roles ¹	✓	✓	✓	✓
Aprobar accesos			✓	✓

Tab. 3.1: Funciones por rol (✓)

Rol / puede asignar	Admin global	Admin local	Creador	Aprobador
Admin global	✓	✓	✓	✓
Admin local		✓	✓	✓
Creador			✓	✓
Aprobador				✓

Tab. 3.2: Asignaciones de roles válidas (✓)

publica, ya que son aspectos guiados por diferentes criterios de seguridad y las personas responsables de los contenidos son quienes deben aprobar los accesos y decidir quienes tienen acceso. La forma en la que los permisos serán asignados de acuerdo a los roles se detalla en la tabla 3.1.

Todos los usuarios pueden asignar a otros usuarios las siguientes reglas de la tabla 3.2:

Estos mecanismos designados, también aplican a la hora de modificar los privilegios de un usuario. En caso de que un usuario vaya a ser desafectado del sistema, debe haber siempre un usuario alternativo como administrador global/local o creador de área de información o contenido. Si en algún momento algún usuario deja de pertenecer a la compañía, y algún área quedara sin administradores, entonces el sistema asignará automáticamente a cargo de los elementos del portal “huérfanos” a quienes sean los responsables inmediatos superiores dentro del mismo, siguiendo la lógica de Aprobador - Creador - Admin Local y Admin Global en última instancia. Para Admins globales debe haber siempre al menos 2 y es parte de la configuración inicial del sistema.

3.3.4. Publicación de contenidos (Creador/Admin).

La publicación de contenidos se hará de modo que cada pieza de contenido estará representada por los siguientes metadatos obligatorios:

- Título: Un texto denomina al contenido.
- Descripción: Texto que da una breve descripción de lo que el contenido ilustra o representa.
- Imagen miniatura: una imagen que representa el contenido, pudiendo ser un logo, un pequeño screenshot o cualquier elemento visual que permita reconocer al contenido publicado.
- Tipo de Contenido: Los contenidos pueden ser de dos tipos, o bien archivos cargables o bien URLs a elementos que residan dentro de la intranet.

- Segmentos de usuarios: Debe haber al menos 1 segmento de usuario (segmento por defecto llamado usuarios generales) y se le deben asignar objetivos de frecuencia de uso, en función de una cantidad de veces por unidades de tiempo. Cantidad: Un número natural mayor a cero. Frecuencia: deberá ser algún valor de esta lista: “diario, semanal, mensual, trimestral, semestral, anual”

De este modo, la idea es poder establecer los objetivos de adopción según perfiles de usuario y cómo se espera que estos utilicen los contenidos para poder ser eficientes en sus procesos de negocio.

- Aprobadores: puede agregarse el listado de aprobadores y por defecto quien crea podrá aprobar acceso.
- Selección del modo de provisión de acceso:
 - En caso que el tipo de Contenido sea un archivo, se aplica la lógica por defecto de acceso directo.
 - En caso de que el tipo de Contenido sea un enlace a un reporte en otra tecnología, se opta entre dos modelos:
 1. Un modelo de control de acceso integrado a un sistema de tickets, donde se debe especificar dentro de ese sistema qué grupo es el encargado de recibir el ticket y se configuran parámetros con los que se creará el ticket, según sean requeridos:
 - Nombre del reporte al que se pide acceso.
 - Nombre de quien solicita el acceso.
 - Handle de usuario.
 - Dirección de email.
 - Puesto en la compañía.
 - País de localización de quien lo pide.
 2. Si el reporte tiene un modelo de acceso basado en listas de distribución del directorio corporativo, entonces deben especificarse dichas listas y quedarán asociadas al contenido (y se pueden asociar a segmentos de usuarios).
 - Si tiene seguridad basada en roles dentro del reporte y de ser así, se definen dos mecanismos posibles:
 1. Listas de distribución específicas del directorio corporativo. En este caso se define qué listas del directorio corporativo están asociadas a este contenido, para que el aprobador pueda utilizarlas en el momento de la aprobación.
 2. Esquema propio de cada contenido con perfiles/roles particulares. En este caso, deben especificarse los end points y credenciales para poder integrar la administración.

3.3.5. Descubrimiento y navegación en el catálogo (Consumidor).

El portal tiene dos modos de operación. Un primer modo predeterminado, que permite utilizar los contenidos asignados, disponibles y ordenados según las unidades geográficas y áreas de información. También hay una sección de “Favoritos” que contiene aquellos contenidos elegidos por el usuario como favoritos, que normalmente simplifican el acceso a los

que son de uso frecuente. Esta la posibilidad de buscar contenidos por textos relacionados a los metadatos definidos en el caso de uso 4. El segundo modo, es el modo “catálogo”, donde el usuario consumidor tendrá la posibilidad de ver por unidades geográficas y áreas de información disponibles pero a los que no tiene acceso. Cada elemento podrá ser agregado a un “carrito de compras”, y una vez terminada la selección de los elementos para solicitar acceso, se podrá hacer un pedido formal de acceso. También está la posibilidad de agregar un texto como solicitud, explicando para cada elemento (o de forma colectiva) por qué la persona necesita acceso para los elementos. Cuando el usuario completa la acción de pedir acceso a los elementos del carrito de compras, se dispararán los procesos de aprobación, otorgamiento de accesos y notificación según se hayan definido en cada elemento.

3.3.6. Circuito de Aprobaciones (Creador/Admin).

En el circuito de aprobaciones, quienes son aprobadores recibirán una notificación por email y también tendrán un ícono en el portal que les indicará que tienen “tareas pendientes”. Tanto el email como el ícono, lleva a los aprobadores a la interfaz que les permite evaluar los pedidos en función de las solicitudes que recibieron y poder aprobarlos o rechazarlos de manera general o particular. En caso de rechazo de la solicitud, deben colocar el motivo y en ambos (tanto positivo como negativo) casos el resultado del proceso será informado al usuario solicitante, tanto por email, como una notificación en el portal. Adicionalmente, si el contenido al que se solicita acceso, está cargado en el portal, el acceso se otorga de modo directo. En caso de que el contenido publicado tenga asociada seguridad manejada por listas de distribución, se hará la asociación en ese momento y si además tiene habilitada la configuración de administración de perfiles de aplicación, se asignarán en el momento. Si por último, el contenido publicado, está implementado y operado bajo un modelo de soporte basado en tickets, con la información configurada en el momento de la creación del contenido en el portal, se generará un ticket mediante el equipo correspondiente, adjuntando la información necesaria para el alta y la aprobación del aprobador para que el equipo de mantenimiento tenga el respaldo necesario para poder documentar y accionar el pedido, de acuerdo a las normas de cumplimiento de la empresa. En este caso, la notificación que recibirá el usuario que solicitó acceso tiene un texto que le informa que su pedido fue aprobado y se creó el ticket nro XXX en su nombre, para que pueda darle seguimiento con el equipo de operaciones.

3.3.7. Navegación, acceso al portal y a reportes y monitoreo de estadísticas.

Las acciones de navegación dentro del portal tienen como objetivo poder recorrer los contenidos y accederlos. Para acceder un contenido, bastará con clicar con el mouse sobre la el ícono que lo representa en el catálogo y esto desplegará un nuevo Frame del navegador de internet que será una dirección enmascarada al elemento en cuestión. Esto permitirá que el enlace se pueda guardar como acceso directo y a la vez compartir por el usuario con otras personas, sin exponer el link original y preservando el control de acceso primario. El efecto deseado es agregar un nivel de indirección de modo tal que al guardar el link como acceso directo, quienes publican los contenidos puedan modificar el link interno de acceso sin afectar los enlaces del portal. Finalmente, todas las acciones de búsqueda, navegación, gestión de usuarios y de accesos, serán registradas en una bitácora de eventos de modo tal de generar un historial de las transacciones ocurridas, para poder luego poder

optimizar el funcionamiento del sitio, aplicar auditorías de cumplimiento necesarias y a la vez obtener métricas que permitan medir los criterios de éxito del Portal.

3.4. Alcance inicial y entregables previstos

El alcance inicial comenzó con la implementación de los casos de uso para los roles de Administrador Global, Creador y Consumidor de contenidos. En una segunda fase se decidió incorporar al perfil de administrador Local para poder agregar una capa intermedia de administración que permita descentralizar y federar el gobierno de los contenidos de modo más eficiente. Se priorizó también la publicación de contenido global en una primera instancia y luego en etapas sucesivas, luego de la implementación del rol local, se comenzó a expandir a otras unidades geográficas con el abordaje a los equipos locales de múltiples áreas.

3.5. Definición de KPIs y métricas de éxito

Para poder medir el avance y el éxito del proyecto se identificaron métricas de varios tipos:

- *Experiencia de usuario*: La primer métrica que se definió está asociada al tiempo de respuesta del portal durante las interacciones y en principio, lo que se buscó es que cada interacción, dure entre 300 y 500 milisegundos (sin incorporar el tiempo de latencia de red). Es decir que cada vez que el usuario disparaba una acción sobre el portal, el tiempo de respuesta combinado, desde que enviaba el estímulo hasta que recibía la respuesta completa (o el indicio de respuesta) no debía ser mayor a 1000 milisegundos. Cuando hablamos del indicio de respuesta, nos referimos a que a veces, por volumen de información que debe viajar desde el servidor de base de datos y web hasta el browser del usuario, simplemente no es posible, pero sí se puede comenzar a renderizar parcialmente o dar alguna indicación visual de que su solicitud fue hecha y que está en proceso de ser resuelta.
- *Adopción*: Para poder medir la adopción efectiva de la herramienta se estableció como métrica la cantidad de piezas de contenido creadas, cantidad de usuarios creadores, cantidad de usuarios asignados y finalmente actividad asociada según los tipos de transacciones definidos en los casos de uso.
- *Ahorros de tiempo y recursos y minimización de errores*: Para poder hacer una medición de estos elementos se decidió hacer seguimiento de cantidad de tickets generados para gestión de accesos a grupos de soporte. Cantidad de accesos asignados mediante listas de distribución.

4. ARQUITECTURA

4.1. Principios de diseño

La arquitectura que se buscó debía contar con los siguientes elementos:

1. Ser una Web-app (3.2.1)

El encarar el proyecto como una aplicación web, vs una arquitectura de cliente servidor tradicional con un cliente ejecutable e instalado en las máquinas clientes y el código del servidor corriendo en un server, permitió aprovechar las ventajas de mantenimiento y escalabilidad propias de las arquitecturas de las aplicaciones web. En esencia, siguen siendo cliente servidor, con la diferencia de que el código que se ejecuta en el cliente se descarga en el código web al que se accede desde el browser de internet, de modo que al mantenimiento y el control de versiones, actualizaciones y mejoras se da de modo automático¹ en cada máquina cliente.

Adicionalmente, con un diseño para dicha arquitectura, adecuado, se puede también escalar horizontalmente sin mayores esfuerzos y aumentar la tolerancia a fallas.

En el caso puntual de esta aplicación, se diseñaron dos configuraciones: una para entornos de desarrollo y pruebas y luego una para un entorno productivo, donde se tolera una carga de usuarios real y debe maximizarse la tolerancia a fallas, donde se agrega un tercer servidor web/de aplicaciones.

En los entornos de desarrollo y de test, siempre se prueba con la siguiente configuración:

Finalmente y de acuerdo a los estándares tecnológicos de este proyecto, se utilizó el siguiente stack tecnológico:

Loadbalancer de F5

Webserver: Apache

Application server: Tomcat

RDBMS: Oracle

Operativo de servers: RH Linux

2. Lean UX (3.2.2)

En el caso del diseño de la interfaz de usuario, no condiciona la arquitectura en sí de la aplicación, aunque influye en la selección de qué tecnologías se utilizarán para implementar las interacciones y los elementos de visualización. La interfaz de usuario, es el sistema de comunicación que tiene el producto de software para brindarle información sobre el estado del sistema, los contenidos y el resultado de las operaciones que el usuario realice con el mismo.

¹ En ciertos casos, es posible que haya código descargado en cachés locales, pero pueden invalidarse y actualizarse fácilmente como parte de las actualizaciones.

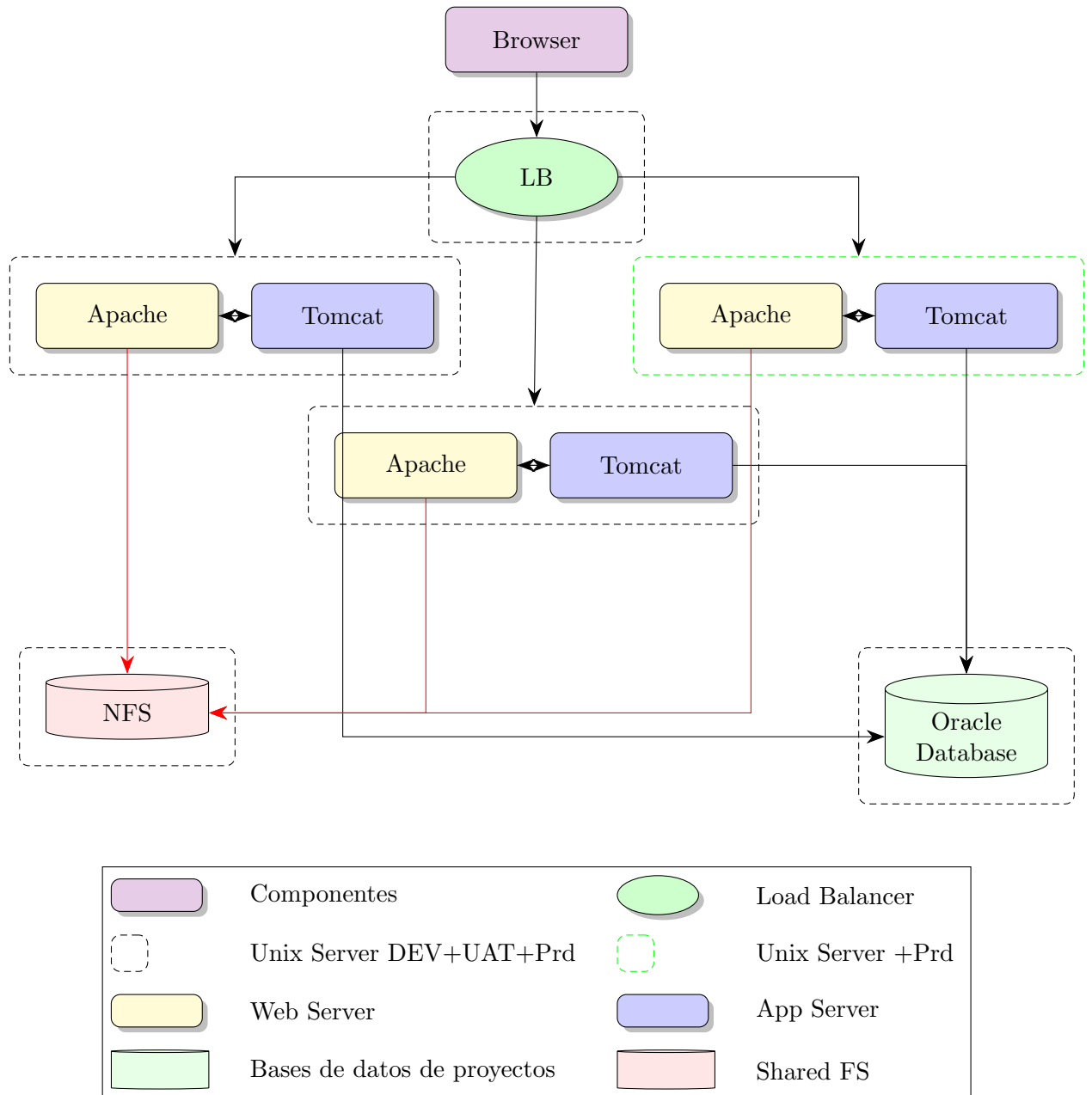


Fig. 4.1: Componentes de la arquitectura de la webapp

3. Tener una SOA (3.2.3)

En este caso particular, se buscó implementar una arquitectura orientada a servicios, para la capa de negocio, de modo que pueda hacerse una separación, escalable de ser necesaria de ciertos servicios críticos y de alta demanda. de este modo, se puede seguir escalando la capacidad de la aplicación, en función de ciertas operaciones de alta demanda y críticas para brindar los servicios del portal. En otros casos, ciertos servicios se definieron como elementos reusables que pueden ser parte de un ecosistema más grande de aplicaciones y que tenía sentido independizar del bloque inicial de código, como proyectos "hijos".

4. Descentralizado y escalable (3.2.4)

La posibilidad de descentralización, tiene que ver con poder distribuir tareas según responsabilidades de ciertos usuarios, en vez de tener un equipo centralizado de gente que opere y administre los contenidos y los manejos de accesos del portal. Esto implica que el volumen de usuarios con la capacidad de ejercer múltiples roles aumenta, en prácticamente un orden de magnitud, de 10x a 100x.

La escalabilidad, se vuelve entonces un aspecto esencial para poder sostener un buen desempeño, desde el punto de vista de la performance y la experiencia de usuario, sin poner en riesgo la estabilidad del sistema. Para dicho fin, la arquitectura de 4.1, mencionada anteriormente se vuelve clave.

5. Auditable (3.2.5) explicar toda la lógica de auditoría en función de qué transacciones que hayan identificar como auditables y luego empezar a desarrollar cada uno, los temas de metadata que deben registrarse. en un log de auditoría se registra:

- userid: identificador unico del usuario que inicia la transacción.
- timestamp (formato de timestamp que representa el instante de tiempo en el que se hizo la transacción con nivel de granularidad de milisegundo).
- operación descripción de la operación que se realiza.
- campos afectados (con valores anteriores y nuevos, en caso de ediciones/actualizaciones).
- id de transacción (para el caso de que haya multiples operaciones en una transacción).
- duracion (en milisegundos).

Con toda esta metainformación, es posible obtener una vitácora de las transacciones ocurridas y poder realizar auditorías y análisis varios, tanto para evaluar estadísticas del sistema como así también patrones de uso y performance.

4.2. Arquitectura

4.3. Componentes principales

Para avanzar con la implementación de la solución, se decidió implementar un patrón de Model View Controller (MVC). Dicho enfoque, permite separar claramente en capas lógicas la presentación e interfaz, la lógica de negocio y el modelo de datos, que se alinean muy bien con la arquitectura de webapp.

El mapeo del modelo MVC a la arquitectura de webapp fue hecha del siguiente modo: Modelo (Model):

En el contexto de una webapp, el modelo representa la parte de la aplicación que maneja los datos y la lógica de negocio. Incluye las clases y objetos que interactúan con la base de datos para almacenar, recuperar y manipular datos. El modelo se comunica directamente con el motor de bases de datos (RDBMS, Oracle, en este caso) para ejecutar consultas y obtener resultados. En esta arquitectura de múltiples servidores, el modelo está distribuido, con diferentes instancias de aplicación (Tomcat) accediendo a la misma base de datos compartida a través de un sistema de archivos compartido (NFS).

Vista (View):

La vista es la parte de la aplicación que el usuario interactúa directamente. En una webapp, es la interfaz de usuario presentada en el navegador del usuario. Las vistas se generan a partir del modelo y son dinámicas, adaptándose a los datos que el modelo proporciona. Las vistas pueden ser renderizadas en el servidor (Tomcat) y/o en el cliente (navegador). En este caso, las vistas son gestionadas por el servidor web (Apache), que sirve archivos estáticos (HTML, CSS, JavaScript albergados en el NFS) y delega las solicitudes dinámicas al servidor de aplicaciones (Tomcat).

Controlador (Controller):

El controlador es el intermediario entre la vista y el modelo. Es responsable de manejar las solicitudes del usuario, interpretar las acciones y actualizar la vista y/o el modelo en consecuencia. Aquí, el controlador se encarga de procesar las solicitudes HTTP, extraer los datos de la solicitud, invocar el código de negocio del modelo y devolver una vista actualizada al cliente. El controlador está implementado en el servidor de aplicaciones, donde maneja las interacciones con el usuario y coordina el flujo de datos entre la vista y el modelo.

Arquitectura de la webapp:

En una arquitectura de webapp basada en MVC, los componentes se distribuyen a lo largo de los diferentes niveles de la stack tecnológico. El servidor web (Apache) actúa como el punto de entrada para las solicitudes y atendiendo las solicitudes estáticas y delegando las dinámicas a los componentes tomcats. El servidor de aplicaciones (Tomcat) ejecuta el controlador y el modelo, procesando las solicitudes y generando las vistas. El sistema de archivos compartido proporciona el soporte al contenido estático que comparten las instancias de Apache y de Tomcats. La base de datos (Oracle) proporciona almacenamiento persistente para los datos generados, mantenidos y accedidos por el modelo.

1. **ReportsHub:** Código que contiene la lógica de negocios y la capa de presentación que se sirve a los browsers con los que se accede.
2. **User Admin API**
3. **RLS API** en este caso se puede explicar como funciona el API de roles para poder integrar como wrapper lo que se decidió implementar.
4. **RH DB** (base de datos de ReportsHub)

Componentes del ecosistema:

1. LDAP

Explicación del directorio de LDAP y las técnicas de integración para poder meterse.

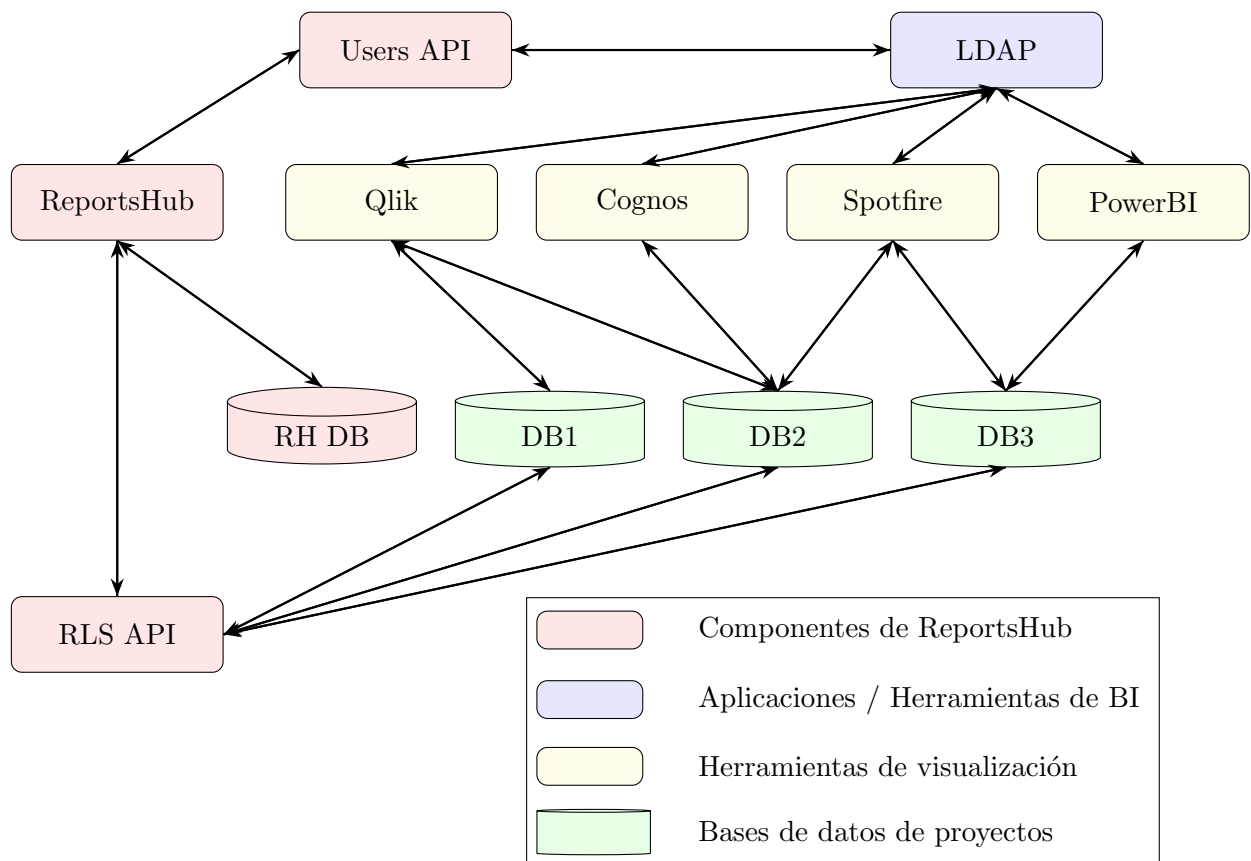


Fig. 4.2: Componentes de Arquitectura

2. herramientas de visualización Aquí uno puede hablar de lo que son las herramientas de visualización y luego profundizar en variaciones.
3. bases de datos de proyectos acá se puede hablar del API

4.4. Diseño de seguridad y Row Level Security (RLS)

Por cuestiones de simplicidad y de diseño general, se trabajó con un esquema de seguridad donde usuarios de ciertas áreas funcionales, pueden tener acceso a la información completa de dichas áreas, con limitaciones geográficas. en el caso de los usuarios "locales", en general se les otorga permiso vinculado al país donde operan. usuarios regionales, se les otorga visibilidad sobre la región (información consolidada a nivel regional) y del detalle de los países incluidos en dicha región. finalmente hay usuarios globales que tienen acceso a una capa global de datos (del área de interés del contenido publicado) y luego a los datos consolidados en niveles geográficos incluidos (regionales y países de cada región).

normlamente para implementar el filtrado a nivel geográfico, se aplicaron técnicas de RLS, de modo que permite mediante el rol del usuario, poder tener scope a las filas (rows) que son relevantes a dicho usuario.

La implementación, normalmente se aplica a nivel de modelo de datos, haciéndola inherente a las consultas, de modo que siempre se agregan automáticamente a todas las queries, where conditions limitando las filas con las que se trabaja por alcance geográfico.

Se puede agregar algún diagrama de RLS y algún modelo de datos (DER) que ayude a implementarlo.

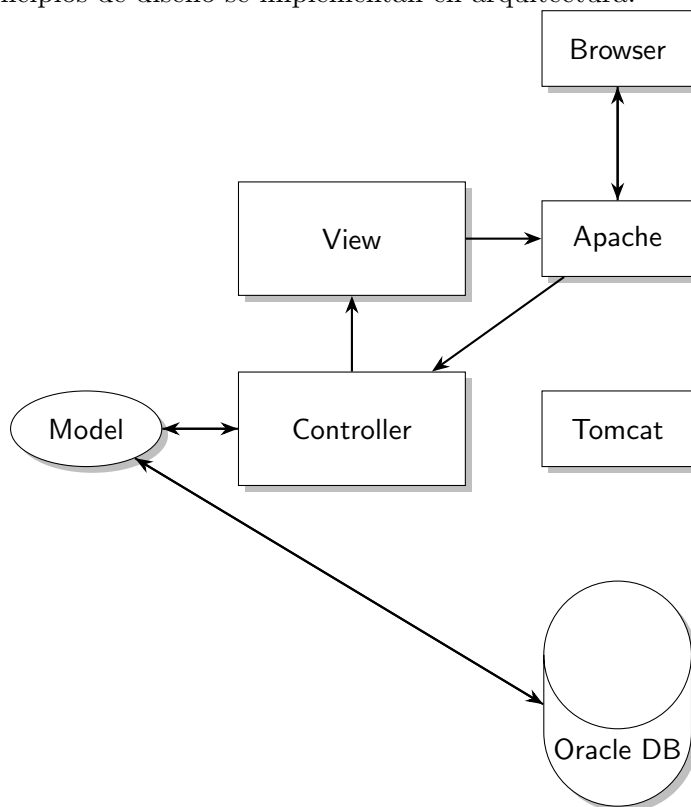
4.5. Flujos de datos e integraciones con sistemas existentes

Workflows de aprobaciones y relación con los casos de uso.

acá se puede trabajar las mejoras de los flujos de interacciones para cada caso de uso.

4.6. Consideraciones de escalabilidad y resiliencia

revisar si acá las condiciones de escalabilidad y resiliencia que se definieron en los principios de diseño se implementan en arquitectura.



5. DESARROLLO Y DESPLIEGUE

5.1. Enfoque metodológico (Waterfall/Agile)

Metodología Waterfall En la metodología Waterfall, las fases del desarrollo de software se llevan a cabo de manera secuencial. Aquí tienes cómo se organiza el equipo en esta metodología:

Product Owner: Responsable de definir los requerimientos y asegurarse de que el producto final cumpla con las expectativas del cliente. **Scrum Master:** Este rol no existe en Waterfall, pero podría haber un jefe de proyecto que coordina las tareas y asegura que las fases se completen a tiempo. **Programadores de Front End:** Se encargan de la interfaz de usuario una vez que se ha definido y aprobado el diseño. **Programadores de Back End y de Base de Datos:** Trabajan en la lógica del servidor y la estructura de la base de datos después de la fase de diseño. **Testers:** Realizan pruebas finales después de que el desarrollo esté completo. **Operadores:** Se encargan del despliegue y mantenimiento del sistema una vez que ha sido probado y aprobado. **Metodología Agile** En la metodología Agile, las fases del desarrollo de software son iterativas e incrementales. El equipo trabaja en sprints cortos y frecuentes para entregar funcionalidades completas. Aquí es cómo se organiza el equipo:

Product Owner: Define y prioriza el backlog del producto, asegurándose de que el equipo trabaje en las tareas de mayor valor. **Scrum Master:** Facilita el proceso Agile y remueve obstáculos para el equipo. **Programadores de Front End:** Trabajan en la interfaz de usuario en cada sprint, colaborando estrechamente con los programadores de back end. **Programadores de Back End y de Base de Datos:** Desarrollan la lógica del servidor y la estructura de la base de datos en cada sprint. **Testers:** Realizan pruebas continuas durante cada sprint, asegurando que las nuevas funcionalidades funcionen correctamente. **Operadores:** Empiezan a involucrarse en el proceso de despliegue más frecuentemente, aunque aún puede haber una separación entre desarrollo y operaciones.

- 5.2. Planificación y backlog inicial
- 5.3. Definición de historias de usuario clave
- 5.4. Iteraciones y sprints principales
- 5.5. Validaciones con usuarios de negocio
- 5.6. Gestión de calidad
 - 5.6.1. Pruebas funcionales
 - 5.6.2. Pruebas de seguridad
 - 5.6.3. Pruebas de performance
 - 5.6.4. Pruebas de integración
- 5.7. Hitos alcanzados y ajustes sobre la marcha
- 5.8. Estrategia de despliegue y fases de rollout.
- 5.9. Piloto inicial y resultados
- 5.10. Escalamiento a más áreas y usuarios
- 5.11. Plan de comunicación y capacitación
- 5.12. Materiales de soporte y guías de usuario
- 5.13. Gestión del cambio organizacional
- 5.14. Mecanismos de soporte post-lanzamiento

6. EVALUACION

- 6.1. Metodología de evaluación**
- 6.2. Resultados cuantitativos**
- 6.3. Resultados cualitativos**
- 6.4. Comparación con KPIs iniciales**
- 6.5. Impacto en seguridad y cumplimiento**
- 6.6. Conclusiones de la evaluación**

7. CONCLUSIONES

- 7.1. Resumen de los logros principales
- 7.2. Relevancia del proyecto para la organización
- 7.3. Impacto estratégico a largo plazo
- 7.4. Próximos pasos sugeridos (roadmap evolutivo)

8. LECCIONES APRENDIDAS

- 8.1. Aspectos que funcionaron bien
- 8.2. Dificultades y cómo se superaron
- 8.3. Factores clave de éxito
- 8.4. Aspectos a mejorar
- 8.5. Aprendizajes organizacionales
- 8.6. Recomendaciones para futuros proyectos

BIBLIOGRAFÍA

- [1] Thomas H. Davenport. «Competing on analytics». En: *Harvard Business Review* 84.1 (2006), págs. 98-107.
- [2] Thomas Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, 2005.
- [3] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison Wesley, 2003.
- [4] W. H. Inmon. *Building the Data Warehouse*. New York, NY: John Wiley & Sons, 1992.
- [5] ISACA. *COBIT 5: A Business Framework for the Governance and Management of Enterprise IT*. Rolling Meadows, IL: ISACA, 2012.
- [6] Ivar Jacobson. «Object-Oriented Development in an Industrial Environment». En: *Proceedings of the OOPSLA '87 Conference*. 1987.
- [7] Glenn E. Krasner y Stephen T. Pope. «A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System». En: *ParcPlace Systems, Inc.* (1988).
- [8] Raymond T. Yeh y Pamela Zave. «Specifying software requirements». En: *Proceedings of the IEEE* 68.9 (1980), págs. 1077-1085.
- [9] Edward Yourdon y Larry L. Constantine. *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. Prentice Hall, 1979.
- [10] Pamela Zave. «A comprehensive approach to requirements problems». En: *Proceedings of the IEEE Computer Society's Third International Computer Software and Applications Conference (COMPSAC '79)*. IEEE, 1979, págs. 117-122.