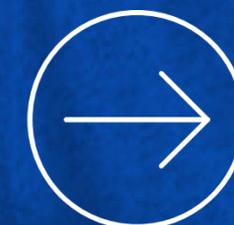


OpenDev^{PRO} Medifé

CASO DE ÉXITO

Unificación de Sistemas de Autorizaciones en Medifé

Simplificando la Gestión de Autorizaciones de Salud



OpenDev^{P.L.O}

Medifé

CLIENTE

Medifé

INDUSTRIA

Salud

DESAFÍO

Medifé enfrentaba la necesidad de utilizar dos sistemas distintos para gestionar las autorizaciones de prácticas de salud en AMBA y en el interior del país. Esta dualidad complicaba los procesos centrales de la empresa, generando ineficiencias y dificultando la gestión unificada de las autorizaciones.

ESTRATEGIA Y EJECUCIÓN EFICAZ

Solución de Opendev Pro

Solución Implementada:

Unificación de Pantallas

Desarrollo de una interfaz unificada para gestionar todas las autorizaciones.

Patrón Backend for Frontend (BFF)

Implementación de BFF para gestionar la obtención de datos de múltiples orígenes de manera eficiente.

Arquitectura de Microservicios

Migración de la lógica de negocio de PL-SQL a aplicaciones independientes utilizando Openshift para escalabilidad.

ESTRATEGIA Y EJECUCIÓN EFICAZ

Solución de Opendev Pro

Solución Implementada:

Integración de Sistemas

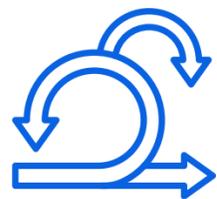
Integración de la información en los dos sistemas Core vigentes de Medifé.

LegacyManager

Garantía de que los sistemas core legacy sigan siendo funcionales durante la transición.

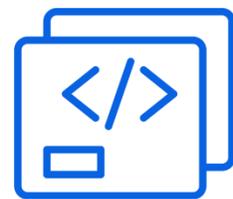
Equipo Especializado

Nuestro equipo experto en el desarrollo del software.



SCRUM MASTER

Garantizando la metodología Agile y comunicación constante.



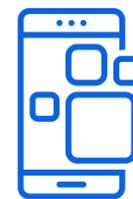
ARQUITECTO

Asegurando la excelencia técnica, escalabilidad y performance.



ANALISTA FUNCIONAL

Definiendo un nuevo workflow adaptado a ambos circuitos.



UX DESIGNER

Optimizando la usabilidad para distintos tipos de autorizaciones.



LÍDER TÉCNICO Y DESARROLLADORES

Creación de la aplicación con Angular (Front End) y Node JS (Back End), integraciones en JAVA.



TESTER

Pruebas unitarias y automatizadas.

CÓMO AYUDAMOS AL CLIENTE

Resultados y beneficios Para Medifé

Impacto Positivo y Transformación en sus procesos

RESULTADOS

<p>Solución de Problemas Eliminación del uso de dos sistemas distintos, simplificando la gestión de autorizaciones.</p>	
<p>Entrega Incremental: Resultados efectivos y medibles en cada sprint, adaptándose rápidamente a las necesidades del cliente.</p>	
<p>Transformación Digital: Implementación de una metodología ágil que permitió una transformación efectiva, con entregas de valor cada 15 días.</p>	
<p>Equipo Robusto: Formación de un equipo con un Scrum Master, arquitecto, analista funcional, UX designer, líder técnico, desarrolladores y tester para una solución integral.</p>	

METODOLOGÍA DE TRABAJO CON MEDIFÉ

Metodología de trabajo y comunicación

Implementación Agile y Colaboración Eficaz



Scrum y Agile

Uso de metodologías ágiles para entregas rápidas y constantes.



Reuniones Diarias

Seguimiento constante y coordinación.



Sprints

Entregas frecuentes de valor cada 15 días



Chapter Meetings

Evaluación y ajustes mensuales.



Comunicación Continua

Transparencia y colaboración con reportes de progreso regulares.



Entregables con el cliente

Entregables:

- **Diagrama Funcional de la Integración:** Descripción detallada de la arquitectura.
- **Documento Funcional:** Especificaciones de los flujos y funcionalidades.
- **Accesos y URLs de Test y Prod:** Información para entornos de prueba y producción.
- **Usuario en la Consola de Monitoreo:** Acceso a la monitorización en tiempo real.
- **Reportes de Avance por Sprint:** Informes periódicos sobre el progreso.

Apartado Técnico-Funcional

Arquitectura de Microservicios

En función a un requerimiento del negocio de la empresa Medifé, el cual se basaba en integrar dos sistemas legacy monolíticos, se diagramó una arquitectura de microservicios. Por políticas de desarrollo del cliente se dispuso la migración de la lógica del negocio de algunos PL-SQL y el mapeo de las tablas a aplicaciones independientes.

En algunos casos específicos y por decisión del cliente se mantuvo el consumo de los "stores" desde un nuevo microservicio. Una vez actualizadas las aplicaciones se integraron en una aplicación cliente, surgiendo la problemática de tener dos orígenes para la obtención de datos. Este desafío lo enfrentamos con el patrón de diseño "Backend for Frontend" (BFF), permitiendo que ante una consulta de una aplicación cliente, sea el BFF el responsable de obtener los datos de los diferentes orígenes (microservicios) abstrayendo a la aplicación que consulta de dicha responsabilidad.

Apartado Técnico-Funcional

LegacyManager

Un punto central en este desarrollo fue el coliving de los sistemas core legacy, ya que mientras eran utilizados desde los microservicios debían seguir siendo funcionales al negocio desde sus core respectivos. Este desafío lo enfrentamos con la implementación de un LegacyManager, aplicación responsable de actualizar en los legacy core lo que sea necesario para que la convivencia entre los sistemas sea óptima hasta que el apagado de los core fuere definitivo.

Plataforma de Contenedores

Para una actualización tecnológica se utilizó Openshift como plataforma de contenedores, lo cual nos permitió escalabilidad en el despliegue de aplicaciones.

Apartado Técnico-Funcional

Api Managers

Se utilizaron diferentes herramientas para dar mayor flexibilidad y solidez a la arquitectura, como por ejemplo Api Managers.

Arquitectura Asíncrona

Este proyecto a su vez demandó una arquitectura más flexible para mejorar la performance dada la gran cantidad de lógica de negocio regida por validaciones y cambios de estado. Para esto en cierto punto se implementó como solución tecnológica una arquitectura asíncrona, basada en la utilización del patrón message-broker, logrando que los microservicios tengan menos carga de trabajo.

¡Gracias!



opendevpro.io



linkedin.com/company/opendevpro

OpenDev_{PRO}

Medifé