



Mis aliados - Suramericana

Pedro Pablo Gallego Pinzon

Informe de práctica para optar al título de Ingeniero de Sistemas

Asesora

Sandra Patricia Zabala Orrego, Especialista (Esp) en Gerencia

Asesor

David Alejandro Gutiérrez Muñoz, Profesional en Mercadeo y Publicidad

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería de Sistemas

Medellín, Antioquia, Colombia

2022

Cita	Pedro Pablo Gallego Pinzon [1]
Referencia	[1] P Gallego Pinzon, “Mis aliados - Suramericana”, Trabajo de grado profesional, Ingeniería de Sistemas, Universidad de Antioquia, Medellín, Antioquia, Colombia, Estilo IEEE (2020) 2022.



Centro de documentación ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/director: Jesús Francisco Vargas Bonilla

Jefe departamento: Diego José Luis Botía Valderrama.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

TABLA DE CONTENIDO

RESUMEN.....	4
INTRODUCCIÓN	5
OBJETIVOS.....	6
General	6
Específicos	6
MARCO TEÓRICO.....	7
METODOLOGÍA	9
RESULTADOS.....	10
REFERENCIAS	15

RESUMEN

En este informe se detalla el trabajo realizado en el proceso de práctica empresarial realizado con la compañía de seguros de vida Suramericana S.A. específicamente el proyecto Mis Aliados, un proyecto con un alto componente tecnológico en el cual se realizaron diferentes acciones de cara a mejorar la calidad del sistema, desarrollar nuevas funcionalidades conforme a los requisitos del negocio y el mejoramiento de la metodología de desarrollo.

INTRODUCCIÓN

El grupo de inversiones suramericana es una empresa fundada en el año 1945 en Medellín - Colombia, está a sido reconocida por consolidarse como una de las empresas más importantes en Latinoamérica por su diversificación en servicios, principalmente orientados a seguros.

Una de las apuestas por las que Sura destino parte de sus utilidades en los últimos años es el Suralab, un conjunto de proyectos de innovación que resultan de la identificación de necesidades en proyectos Core de la empresa y que deben desarrollarse de una forma más cercana a emprendimientos propios.

Es así como surge el proyecto Mis aliados, que tiene como objetivo buscar la estabilidad laboral de trabajadores independientes a través de una plataforma que permite conectar personas con necesidades de un servicio en específico con independientes (aliados) que puedan realizar dichos servicios.

Mediante este proyecto se identificarán las problemáticas que los usuarios (contratantes e independientes) y desarrolladores pueden estar presentando en el sistema para realizar sus actividades diarias y así mismo se generará un plan de mejoras que permita dar solución a cada uno de estos problemas anteriormente planteados.

OBJETIVOS

General

Implementar mejoras en el sistema del proyecto mis aliados que permita aumentar la calidad y efectividad de los cambios o actualizaciones de este, desarrollando nuevas funcionalidades que el negocio requiera y contribuir a la estabilidad de este sistema.

Específicos

- Verificar el estado actual del proyecto investigando en la documentación de este, sobre las tecnologías utilizadas y como es el flujo de desarrollo del equipo.
- Identificar oportunidades de mejora implementando nuevas tecnologías.
- Implementar mejoras usando nuevas tecnologías solventando cada una de las oportunidades de mejora identificadas.

MARCO TEÓRICO

El sistema inició su desarrollo aproximadamente en el año 2017, en su momento se eligió un stack de tecnologías compuesto por una base de datos relacional MySQL, backend Java con el framework Spring Boot y frontend Javascript con el framework Angular JS para el desarrollo de la capa visual del sistema.

La arquitectura que se tiene en este momento para el sistema es la siguiente:

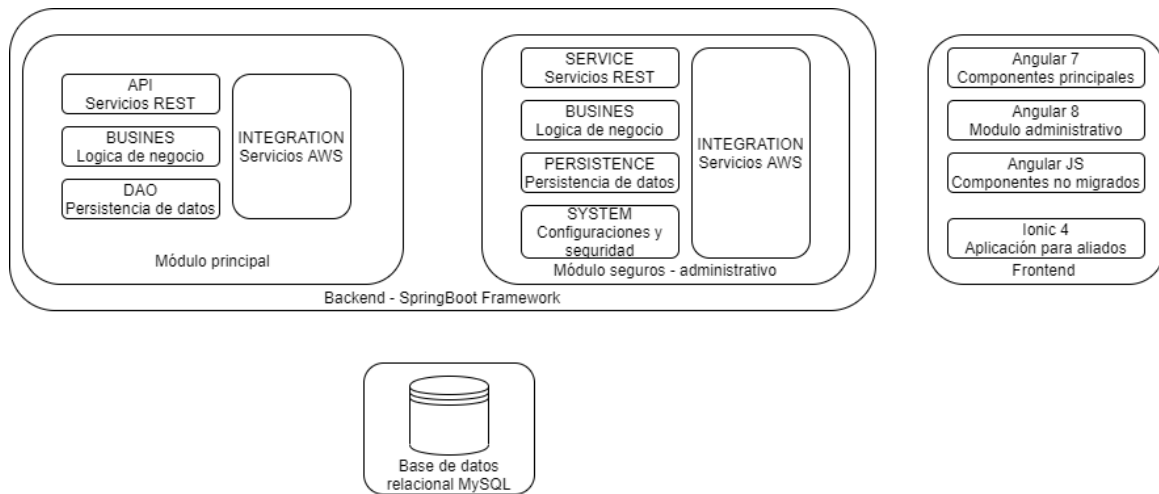


Figura 01. Arquitectura Mis aliados.

Angular es un framework de código abierto mantenido por Google para el desarrollo de aplicaciones frontend modernas, de complejidad media o elevada, permite crear aplicaciones tipo SPA (Single Page Application) PWA (Progressive Web App), ofrece una base para el desarrollo de aplicaciones robustas, escalables y optimizadas, que promueve además las mejores prácticas y un estilo de codificación homogéneo y de gran modularidad [1].

Spring framework es una plataforma de código abierto bajo la licencia Apache[2] que ofrece ayudas al desarrollador a la hora de desarrollar, ya que simplifica el uso de patrones de desarrollo permitiendo aplicar buenas técnicas de programación y dando prioridad a las funcionalidades del negocio, algunas de sus características más importantes son[3]:

- Inyección de Dependencias
- Eventos, recursos, i18n, validación, enlace de datos, conversión de tipo, SpEL.
- Acceso a datos: soporte DAO, JDBC, ORM, Marshalling XML.
- Gestión de transacciones.

- Integración: comunicación remota, JMS, JCA, JMX, correo electrónico, tareas, programación, caché.
- Pruebas (Testing): simulacro de objetos, el framework TestContext, Spring MVC prueba, WebTestClient.
- Programación orientada a aspectos (AOP): permite la implementación de rutinas transversales.
- MVC (Modelo Vista Controlador).
- Seguridad.
- Frameworks web: Spring WebFlux y Spring MVC.
- Procesamiento de datos por lotes.
- Administración Remota: a través de este módulo se puede configurar la visibilidad y gestión de los objetos Java para la configuración local o remota vía JMX.
- Es un framework liviano debido a su implementación POJO (Plain Old Java Object), Spring Framework no obliga al programador a heredar ninguna clase ni a implementar ninguna interfaz.

METODOLOGÍA

Inicialmente se realizó un trabajo de diagnóstico a cada uno de los componentes del sistema de forma transversal así:

- Herramientas para la gestión del proyecto y comunicación en el equipo (Git, Jira...).
- Arquitectura del sistema (lenguajes, Frameworks, gestores de dependencias) y sus versiones.
- Infraestructura.

Luego de finalizada la fase de diagnóstico se procedió a identificar oportunidades de mejora, comunicar estas recomendaciones a los dueños del producto para que estos gestionaran la prioridad en el backlog de desarrollo del proyecto.

Finalmente se procedió a iniciar con la implementación de cada una de las mejoras al sistema segmentadas en diferentes épicas:

- Experiencia de usuario (nuevas funcionalidades).
- Rendimiento y escalabilidad.
- Seguridad.
- Monitoreo.
- Deuda técnica.

RESULTADOS

Diagnostico:

- Herramientas para la gestión del proyecto:
 - Gestión del versionamiento: Gitbucket Suramericana, herramienta básica de Git con limitaciones en sus funciones, solamente accesible desde la red de Sura y en una interfaz poco usable.
 - Gestión del proyecto: Jira Suramericana.
 - Comunicación en el equipo: Whatsapp personal.
- Arquitectura del sistema:
 - Frameworks:
 - Angular JS: Componentes heredados con poca o nula posibilidad de cambio.
 - Angular 7: Nuevos componentes, escalable y una estructura del proyecto basada en componentes.
 - Ionic 4: Deuda técnica evidenciable, estructura del proyecto poco definida.
 - SpringBoot 1.5: Deuda técnica evidenciable, estructura del proyecto poco definida, nula cobertura de pruebas, anti-patrones de desarrollo detectados.
 - Gestión de dependencias:
 - NPM: Dependencias con advertencias de obsolescencia o vulnerabilidades de seguridad.
 - Gradle: Dependencias poco actualizadas, documentadas, poco aprovechamiento del Framework principal (Springboot) agregando otras dependencias que realizan funcionalidades ya cubiertas por este.
- Infraestructura:
 - Arquitectura 100% en nube.
 - Despliegues manuales.
 - Utilización parcial de CDN como mecanismo de distribución local de contenido.
 - Monitoreo reactivo basado en logs.
 - Persistencia de datos basada en base de datos relacional (MySQL).
 - Base de datos accesible de a través de internet.
 - VPC compartida para los ambientes de desarrollo, pruebas y producción.

- Seguridad: Pruebas de seguridad por parte de un tercero arrojaron vulnerabilidades importantes (resultados confidenciales).
- Rendimiento: No se encontró documentación de pruebas de rendimiento al sistema.

Implementaciones realizadas: En el equipo de desarrollo, luego de su priorización por parte de los dueños de producto, se logró la implementación de mejoras segmentadas en las épicas:

- Herramientas para la gestión del proyecto:
 - o Integración continua Jenkins Suramericana para la construcción y despliegue del sistema en todos sus módulos.
 - o Gestión del versionamiento: Bitbucket, mejora la usabilidad para la gestión de pull request, disponibilidad (tecnología en nube).
 - o Integración continua Code Pipeline AWS para la construcción y despliegue del sistema en todos sus módulos (Luego del cambio en el gestor del versionamiento fue necesario migrar de tecnología para el despliegue continuo debido a políticas de Sura).
 - o Adopción de canal de comunicación Slack e integración con gestor de versionamiento, calendario, alertas de despliegue continuo y construcción del proyecto.
- Experiencia de usuario (nuevas funcionalidades).
 - o Desarrollo panel administrativo: Permite al equipo de procesos y atención al cliente realizar las siguientes funciones:
 - Creación, administración, envío de pólizas y sus caratulas a los clientes.
 - Creación y administración de usuarios (contratantes e independientes).
 - Creación y administración de solicitudes y cotizaciones.
 - o Desarrollo nueva plataforma contratantes: El equipo de experiencia diseño un flujo para que los contratantes realicen las solicitudes de servicios y compra de seguros a través de la plataforma con nuevas interfaces y reglas de negocio (<https://misaliados.com.co/>).
 - o Despliegue Frontend independientes en su versión web: permite a los independientes que no cuentan con dispositivos Android acceder al sistema a través de una aplicación Web (<https://aliados.misaliados.com.co/>).
- Rendimiento y escalabilidad.

- Implementación de nueva infraestructura:
 - Balanceador de carga recomendado (ALB) en Subnet de acceso público.
 - VPC aislada solo con los componentes del ambiente de producción.
 - Instancias en Subnet privada, únicamente accesibles por el balanceador de carga.
 - Base de datos aislada en la VPC únicamente accesible por las instancias que se encuentran en la misma Subnet privada.
 - Implementación instancia Bastion para el acceso a la base de datos de producción.
 - Implementación de Cache Redis (ElastiCache) para el acceso a datos en memoria.
 - Habilitación de CDN (CloudFront) para todo el contenido estático (imágenes, videos...).
- Implementación y ejecución de pruebas de rendimiento (Jmeter) a la nueva infraestructura comprobando capacidad de auto escalamiento. (resultados confidenciales).
- Seguridad:
 - Se delego la seguridad del sistema a Firebase Auth con una implementación basada en JWT y Spring Security.
 - Se implementaron mejoras a la seguridad basadas en los resultados arrojados por el informe de seguridad realizado por un tercero. (resultados confidenciales).
- Monitoreo:
 - Implementación de logs de auditoria para las transacciones del sistema (CloudWatch).
 - Implementación de logs de errores con información necesaria para la gestión de incidentes (cuerpo de la petición REST, StackTrace de la excepción...)
 - Implementación de monitoreo Proactivo basado en alertas integradas a través de un WeebHook de Slack que alerta en tiempo real al equipo de desarrollo en caso de excepciones de seguridad o excepciones no controladas con información necesaria para la gestión de incidentes (cuerpo de la petición REST, StackTrace de la excepción, usuario en sesión...).

- Deuda técnica:
 - Cambio en el gestor de dependencias de Gradle a Maven.
 - Actualización y depuración de dependencias (se listan las más importantes):
 - SpringBoot 2.3.1 (última versión estable).
 - SDK AWS 2.X.X (última versión estable).
 - Implementación de patrones de desarrollo y estándares S.O.L.I.D.
 - Eliminación del 100% de Advertencias en el Backend arrojados por el IDE.
 - Análisis de código estático Sonar y reducción del 53% en sus advertencias.

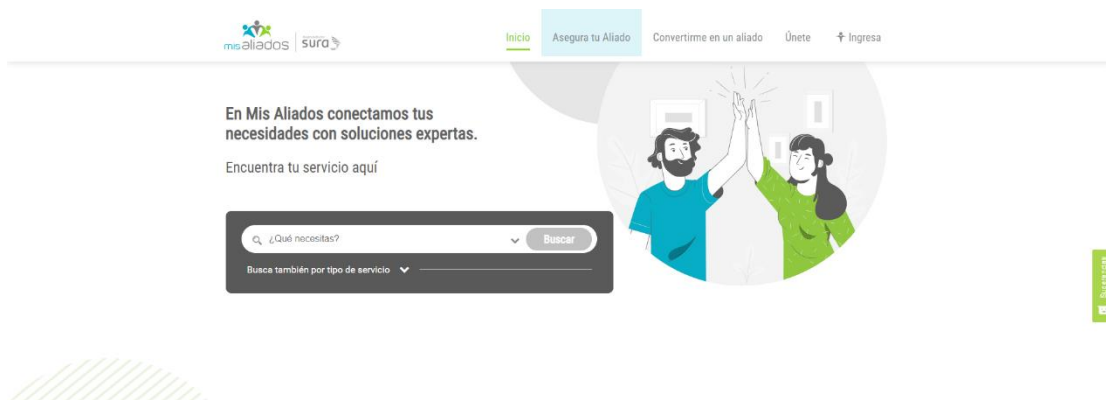


Figura 02. Aplicación web Mis aliados para contratantes ([Enlace](#)).



Figura 03. Aplicación web Mis aliados para independientes ([Enlace](#))

CONCLUSIONES

El proyecto Mis aliados ha tenido una evolución importante a lo largo del trayecto desde que se inició el proceso de práctica empresarial, lo cual ha incidido en nuevos requerimientos funcionales y no funcionales para el equipo de tecnología, así como también impone nuevos retos que permitan superar brechas técnicas en cuanto al rendimiento y seguridad del sistema, con el fin de ponerlo a punto y este pueda ser una solución para conectar contratantes con independientes en Colombia.

Como se evidencio en la fase de diagnóstico, en el proyecto se contaba desde sus inicios con una plataforma con abundante deuda técnica, lo cual inicialmente fue un impedimento a la hora de iniciar de lleno con el desarrollo de nuevas funcionalidades, es necesario concertar con el negocio y dejar claro que la calidad en el software debe ser un requisito que se toma en cuenta a la hora de estimar, con el fin de que no se convierta en deuda técnica que como se vivió en el proyecto requirió incluso en reconstruir el núcleo del sistema.

Definir desde el inicio del proyecto una arquitectura para este y que todo el equipo de desarrollo la conozca y respete permite tener un sistema con calidad y que sea escalable en el tiempo, es importante tener el pensamiento de escalabilidad y pensamiento a futuro sobre las funcionalidades que se desarrollan en el presente y tener claro la volatilidad que puede tener un sistema según su naturaleza, en el caso de Mis aliados, al ser un proyecto de innovación que día a día debe estar probando y experimentando nuevas experiencias para los clientes requiere que el sistema sea adaptable a cambios.

El uso de herramientas de desarrollo es una decisión que no debe tomarse a la ligera, debe definirse en conjunto con todo el equipo un flujo de desarrollo que incluya tareas de revisión por pares, aprobación por parte del negocio de nuevas funcionalidades y técnicas de despliegue continuo que busquen evitar reprocesos e incidentes en producción, logrando así tener un sistema estable y de alta calidad.

La metodología de desarrollo debe ser de conocimiento de todo el equipo en el proyecto y no solo del equipo de tecnología, realizar actividades conjuntas para tener una clara definición de las historias de usuario, requisitos funcionales y no funcionales que debe cumplir el sistema y que impacten en la estimación en tiempo y recurso de cualquier tipo de desarrollo permiten mantener una comunicación fluida de todo el equipo y que el negocio entienda desde lo funcional que implica optar por realizar un desarrollo

REFERENCIAS

- [1] Anónimo, "Angular", 2013. [En línea]. Disponible en: <https://desarrolloweb.com/home/angular>. [Accedido: 12- Jul- 2020].
- [2] Sam Brannen, "spring-framework - license.txt". [En línea]. Disponible en: <https://github.com/spring-projects/spring-framework/blob/master/src/docs/dist/license.txt> [Accedido: 12- Jul- 2020].
- [3] Muradas M, Yanina, "Conoce qué es Spring Framework y por qué usarlo", openwebinars.net, 2018. [En línea]. Disponible en: <https://openwebinars.net/blog/conoce-que-es-spring-framework-y-por-que-usarlo/>. [Accedido: 12- Jul- 2020].