PRÁCTICA EMPRESARIAL
GESTIÓN UTILITARIOS PARA LA COMPAÑÍA SURAMÉRICA S.A
(AUTO GESTIÓN DE CLIENTES)

YESID MONTOYA PRECIADO

ASESORES:
ANDRES MARIN LOPERA
MARIO ALEJANDRO ARISTIZABAL ALZATE

UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
MEDELLÍN
2018

1 . TÍTULO

Gestión Utilitarios para la compañía Suramérica S.A

2. Resumen

En la compañía se tenían una cantidad de requerimientos de cargas masivas los cuales eran asignados a diferentes analistas, quienes dedicaban gran parte de su valioso tiempo a la atención de estos requerimientos por medio de Querys o Stored Procedures, dado que el tiempo empleado por los analistas para darle solución a estos diferentes requerimientos era demasiado, se pensó en crear una aplicación que permitiera darle una cara (Front End) a cada Query o Store Procedure, que fuera de fácil interacción para los usuarios que solicitan los requerimientos y así facilitarles la ejecución de estos requerimientos, para evitar el gasto de tiempo de los analistas en estas soluciones.

Dado el requerimiento de esta aplicación se procedió a su creación pensando inicialmente que fuera de fácil configuración y dinámica con los diferentes parámetros que podrían ser ingresados en un Store Procedure; inicialmente se implementó el requerimiento que más se estaba reiterando en los últimos meses, el cual era una carga de traspasos masivos de los clientes de la compañía que tenían una maduración de documento, es decir que pasaban de tener un tipo de documento a otro por su cumplimiento de edad, actualmente la aplicación solo cuenta con esta operación tanto de manera masiva como de manera unitaria.

3. Introducción

En la compañía se estaba realizando una serie de cargas de información de manera poco productiva, ya que los usuarios de ciertas aplicaciones requerían del apoyo de un analista para esta operación, con el objetivo de agilizar y optimizar estas cargas de información se llevó a cabo el desarrollo de la aplicación web Auto Gestión de Clientes, la cual permite a los usuarios de las aplicaciones realizar estas cargas de forma autónoma mejorando la productividad y el servicio al cliente de la compañía.

La aplicación cuenta con dos modos de carga de información, el primero de ellos será el modo Gestión uno a uno, el cual permite agregar solamente un registro a una de las tablas de la base de datos seleccionada. El segundo permite hacer una carga masiva de registro por medio de un archivo en exel que contenga la información a cargar en un formato específico.

Para el desarrollo de la aplicación se contó con el apoyo de un analista desarrollador de la empresa proveedora pratech en conjunto con el apoyo del practicante Yesid Montoya Preciado, el cual fue el encargado de gestionar el proyecto durante su desarrollo.

4. Objetivos

4.1 Objetivo General.

- Agilizar y optimizar la carga de información de los diferentes aplicativos de la compañía.

4.2 Objetivos específicos

- Mejorar la atención al cliente.
- Garantizar la calidad de los datos de la compañía.
- Alivianar cargas y responsabilidades a los analistas de la compañía.
- Agilizar la atención de asuntos claves de la compañía.

5. Marco Teórico

Dados los beneficios y oportunidades que ofrece una aplicación web, se desarrolló este proyecto entorno a este concepto, una aplicación web está formada por un sistema informático en donde los usuarios la utilizan de tal forma que acceden a un server web a través de Internet o de una intranet, este tipo de aplicaciones son reconocidas por la practicidad del navegador web como cliente ligero.

La popularidad de Internet en el mundo, abrió infinidad de posibilidades en cuanto al acceso a la información desde cualquier lugar o sitio Web. Esto implica un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar la Web de una manera funcional, por esto se decidió realizar una aplicación auto contenida en un contenedor docker en lugar de una aplicación en un servidor de aplicaciones WebLogic.

Además, hoy en día es muy común el uso de dispositivos móviles, con los cuales es posible es acceso a aplicaciones web a través del navegador del dispositivo, esto nos lleva a otro concepto conocido como diseño web responsivo, el cual es una técnica de diseño web que busca la correcta visualización de una misma aplicación en distintos dispositivos. Desde ordenadores de escritorio a tabletas y móviles.

Hoy en día accedemos a sitios web desde todo tipo de dispositivos; ordenador, Tablet, Smartphone… por lo que, cada vez más, nos surge la necesidad de que nuestra web se adapte a los diferentes tamaños de los mismos.

Pero más allá de ser responsiva una aplicación web debe poseer una buena conexión a base de datos para garantizar su alta disponibilidad y evitar retrasos, una base de datos se puede explicar como un acumulado de datos exhaustivo no redundante que están estructurados y organizados de una forma independiente de su utilización e implementación en máquinas accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente, por esto es muy importar que la conexión a esta base de datos sea lo más estable y eficiente posible, para que los diferentes usuarios que utilicen la aplicación lo puedan hacer al mismo tiempo de manera óptima y segura.

5.1. Detalle de la necesidad

Se requiere desarrollar un aplicativo web independiente para la gestión dinámica de tablas de bases de datos de sura. La necesidad está enmarcada dentro de las funcionalidades presentadas en el prototipo gráfico que se encuentra en el siguiente link:https://marvelapp.com/2d333c5/screen/38613779

5.1.1. Carga uno a uno:

- Podrá seleccionar la base de datos
- Podrá seleccionar el esquema asociado a la BD
- Podrá seleccionar la tabla relacionada al esquema
- Podrá seleccionar los “store procedure” válidos para cada acción presentada en pantalla
- Renderizar dinámicamente la tabla seleccionada en la pantalla
- Permitirá tener filtros por los campos que los SP de Sura permitan.
- Acciones:
  - ❖ Crear registro: llama SP provisto por Sura
  - ❖ Eliminar registro: llama SP provisto por Sura
  - ❖ Editar registro: llama SP provisto por Sura
- Si se desea tener validaciones en el front, sura deberá proveer SP con la metadata de las tablas a renderizar, para dinámicamente aplicar las restricciones.
- Se deberá tener un modelo de datos propios de la aplicación para parametrizar las condiciones básicas del aplicativo (conexiones, bases de datos, esquemas, tablas, alias, etc.)

5.1.2. Carga masiva:

- Podrá seleccionar la base de datos
- Podrá seleccionar el esquema asociado a la BD
- Podrá seleccionar la tabla relacionada al esquema
- Podrá seleccionar los “store procedure” válidos para cada acción presentada en pantalla
- Podrá cargar un archivo en formato csv el cual deberá corresponder a la tabla seleccionada
- Para procesar el archivo, se llamarán a los sp dispuestos por Sura para generar las acciones sobre la tabla seleccionada
- Deberá mostrar en pantalla el número de registros cargado correctamente
- Deberá mostrar en pantalla el registro de errores ocurridos al tratar de ejecutar las acciones sobre la BD.

5.2. Tecnologías:

- Java 1.7 o superior
- Contenedor Docker
- Angular 5
- Integración continua
- Perfilación por Seus 4

- Registro de logs en Splunk

6. Metodología

El desarrollo del proyecto se llevó a cabo con el apoyo de las siguientes metodologías ágiles, para garantizar que se logre un excelente resultado en tres pilares fundamentales: Productividad, Calidad y Felicidad.

6.1. Metodología Scrum

Scrum es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software. La metodología indica pone al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan.

Scrum valora por encima de cualquier cosa:

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

6.2. Metodología Kanban

Kanban es una palabra japonesa que significa algo así como “tarjetas visuales” (kan significa visual, y ban tarjeta). Esta técnica se creó en Toyota, y se utiliza para controlar el avance del trabajo, en el contexto de una línea de producción.

Las principales reglas de Kanban son las tres siguientes:

- Visualizar el trabajo y las fases del ciclo de producción o flujo de trabajo
- Determinar el límite de “trabajo en curso” (o Work In Progress)
- Medir el tiempo en completar una tarea (lo que se conoce como “lead time”).

6.3. Desarrollo guiado por pruebas (TDD)

El Test Driven Development (TDD) o “Desarrollo guiado por pruebas” es un enfoque de desarrollo ágil de software en el que primero se escriben las pruebas y luego el código necesario para que la prueba sea exitosa.

Ventajas:

- Mayor calidad
- Diseño enfocado en las necesidades
- Mayor simplicidad en el diseño
- El diseño se va adaptando al entendimiento del problema
- Mayor productividad
- Menos tiempo invertido en debugging de errores

6.4. Automatización de pruebas

Consiste en el uso de software especial para controlar la ejecución de pruebas y la comparación entre los resultados obtenidos y los resultados esperados. La automatización de pruebas permite incluir pruebas repetitivas y necesarias dentro de un proceso formal de pruebas ya existente o bien adicionar pruebas cuya ejecución manual resultaría difícil.

Ventajas:

- Rapidez: Las herramientas de pruebas automatizadas corren significativamente más rápido que los analistas de pruebas humanos.
- Fiabilidad: Las pruebas ejecutan precisamente las mismas operaciones cada vez que se ejecutan, eliminando el error humano.
- Repetición: Se puede probar cómo reacciona el software bajo repetidas ejecuciones de las mismas operaciones.
- Programable: Se pueden programar pruebas sofisticadas y complejas que muestran información oculta de la aplicación.
- Reusabilidad: Se pueden re-utilizar los scripts con pruebas automatizadas, las funciones, los métodos etc…

7. Resultados y análisis

Al día de hoy se han invertido un total de 330 horas en la ejecución del proyecto, inicialmente se tenía una bolsa designada de 300 horas para su ejecución total, debido a que se han tenido problemas con el despliegue en el ambiente de desarrollo esto ha tomado más tiempo del que se esperaba, por lo cual se asignaron 60 horas más a la ejecución del proyecto, a continuación, se tiene el informe de las horas invertidas en la ejecución del proyecto.

REPORTE EJECUCIÓN - AUTOGESTIÓN DE CLIENTE - SURA				
Sprint	**Nombre de la Historia de Usuario**	**Nombre de la tarea**	**Tiempo ejecutado**	**Comentarios**
1		Gestión y seguimiento de proyecto	31.08	En curso la certificación de QA
	Montaje Estructura App SURA	Contextualización solución propuesta a equipo	68	
		Creación repositorio fuentes en GitSura		

		Montar aplicación HolaMundo con la plantilla de arquitectura Sura		
		Montar ambiente Weblogic 12.1.3 local		
		Crear clase para leer sp´s		
		Crear clase para leer modelos		
		Crear servicio para leer los modelos parametrizados		
		Crear servicio para envió de modelos a ejecutar en back		
		Crear clase para procesar los modelos recibidos		
		Integración Frontend - Backend		
		Crear artefacto para pintar mensajes de alertas dinámico de manera global		
		Montar frontend en plantilla UI sura		
		Montar Backend en plantilla Back sura		
	Pintar Opciones Dinámicamente	Validar lista de recursos de opciones dinámicas	13.25	

	Construir servicios base para validar los recursos de las opciones		
	Construir modelo backend		
Gestionar datos uno a uno	Creación repositorio y proyecto en BitBucket	49	
	Header Interfaz		
	SideBar Interfaz		
	Maquetación - Gestión uno a uno		
	Construir formulario dinámico para renderizar campos parametrizados		
	Configuración json de configuración de campos		
	Configuración json de configuración de listas		
	Diseño Loading		
	Diseño Opciones de usuario (Cerrar sesión)		
	Construir pruebas unitarias para el controlador		
	Construir pruebas unitarias para los		

		métodos que se implementan		
		Refactorización general del módulo		
	Agregar opciones en el menú dinámico	Configuración de json para opción de menú	23	
		Leer argumentos configurados para renderización		
		Construir modelo backend		
		Construir formulario dinámico para renderizar opciones de menú		
		Menú de navegación con submenu		
2	Gestionar datos masivamente	Diseño Carga masiva	56	
		Construir modulo masivo en el frontend		
		Construir servicio rest para recibir el archivo de carga		
		Construir servicio angular para envió de archivo		
		Construir clase para validar y procesar archivo de carga		

		Construir clase para recibir los datos procesados		
		Construir DAO para ejecución de datos masivo		
		Construir pruebas unitarias para el controlador		
		Construir pruebas unitarias para los métodos que se implementan		
		Refactorización general del módulo		
		Integración con diseño para ajustes del módulo front		
		Construir método front para procesar campos dependientes		
		Agregar al json campos dependientes		
		Configuración de datasoruce genérico desde json		
		Leer datasource desde el json		
		Crear método para gestionar conexiones a partir del json		
		Modificar método para ejecutar procedimientos almacenados		

Autenticar Usuarios	Activar los interceptors del frontend para validar la seguridad	11.08	
	Configurar header de la app con los datos de seus		
	Configurar el servicio de salir app		
	Configurar el servicio de cerrar sesión app		
	Obtener dni del usuario autenticado		
	Configurar seus ambiente local con todas las urls de la app	1	
Mostrar Informes de Fallas	Modelar clase DTO para el envío de la información de respuesta	14	
	Construir método front para procesar el DTO de respuesta		
	Construir modelo frontend para procesar en el método de reportes		
	Refactorización general del módulo		
Actividades transversales para cumplir con devops	Embeber angular en back	17	
	Datasource dinámicos		

	Instalar la app en ambiente desarrollo sura	3	
	Montar los fuentes desarrollados al sonar (Pratech) para poder cumplir con devops	6	
	Test unitarios de los fuentes que no desarrollamos nosotros (Pratech) para poder cumplir con devops	8	
	Montar app al jenkins de sura	4	
	Montar app en Jenkins Sura	7	
	Acompañamiento Arquitectura	2	
	Reducción deuda técnica Sonar	16	

Horas compradas	360
Horas ejecutadas	329.41
Horas por ejecutar	30.59

Reporte: Hasta 5 de Diciembre de 2018	

Dentro de las actividades más relevantes se tiene la modificación de los Store Procedure (SP) para que puedan ser llamados desde la aplicación sin que afecte el funcionamiento de las bases de datos de la compañía y así garantizar la calidad de la información. Esta modificación consistió en dejar los SP guardando en una tabla temporal, para que posteriormente en horarios de menos tráfico un Job tome los registros almacenados en esta tabla con estado PROGRAMADO y ejecute la lógica necesaria.

Esta modificación fue necesaria dado el tiempo que tomaba cada carga y el número de conexiones a base de datos que dejaba abiertas esperando que la aplicación terminará de ejecutar los SP, con esta mejora se disminuyó el tiempo de respuesta de la aplicación y se garantiza la disponibilidad de la misma.

Dado este trabajo se procedió con el diseño del Front En de la aplicación teniendo en cuenta los parámetros que reciben los procedimientos almacenados (SP), este diseño inicial fue basado completamente en estos parámetros según la necesidad inmediata, pero el diseño fue creado de forma dinámica, es decir que si en otro proceso del negocio se requieren de más o menos parámetros estos puedan ser pintados fácilmente con una pequeña modificación en el código de la aplicación. En el Anexo 1 se observa como fue el diseño para la necesidad que se tenía que era la de traspasos de clientes.

Actualmente la aplicación se tiene diseñada y lista en ambiente local; se han tenido problemas para hacer el despliegue en el ambiente de desarrollo debido a que es una aplicación auto contenida desarrollada bajo un modelo proporcionado por la compañía el cual tiene problemas con el loguer propio, se han llevado a cabo varias reuniones con el área de arquitectura y no se ha podido determinar la causa del problema en nuestro proyecto, ya que en otros despliegues no había presentado este tipo de problema, procedimos a realizar los descartes recomendados por el área y se sigue presentando el problema. En el Anexo 2 se evidencia el error que se está presentando. Dado este fallo no se ha podido desplegar ni probar la aplicación con datos reales, por lo cual no estamos seguros de que esté funcionando como se espera.

Los resultados obtenidos no han sido los más satisfactorios, ya que lo ideal fuera que al día de hoy la aplicación ya debería estar probada y funcionando en ambiente de producción, sin embargo, dados los problemas de despliegue no se han logrado los objetivos esperados.

En el transcurso de mi labor como practicante dentro de la compañía he desarrollado otras actividades importantes como el soporte de aplicaciones claves para la gestión de los clientes, para este soporte se han venido desarrollando trabajos de recolección de información para darle un enfoque orientado a la eliminación del soporte que más se esté presentando, al día de hoy se han gestionado más de 1.800 soportes donde se ha identificado la aplicación que más problemas está ocasionando y así se han podido atacar estos puntos críticos para mantener la menor cantidad posible de incidentes.

Este trabajo ha contribuido satisfactoriamente al equipo bajando el número de personas requeridas para la atención de estos soportes, el día de mi llegada al equipo se tenían 3

personas casi 100% dedicadas a estos temas, el día de hoy solo se tiene una persona enfocada en esto y no requiere del 100% de su tiempo.

8. Conclusiones

Dado que el proyecto no se ha podido dar por finalizado por errores en el despliegue en ambiente de desarrollo es evidente que la implementación de las nuevas tecnologías en ambientes donde no se tiene mucho conocimiento pueden hacer que los proyectos tomen un poco más de tiempo de lo planteado, aunque ha tomado más tiempo de lo esperado aún no hemos dado como un fracaso el proyecto dado que actualmente cuento con contrato hasta el próximo 16 de Julio y la principal meta es cumplir a cabalidad con los objetivos del proyecto antes de dejar la compañía.

La meta es tener el proyecto en producción para el primero de marzo, dependemos de los analistas del área de arquitectura para darle solución al actual problema de despliegue, ya que el programa fuente a partir del cual se empezó con el desarrollo de la aplicación fue proporcionado por ellos y es el que nos está causando inconsistencias con el logger que trae implementado por defecto (Sura Logger).

Como estudiante fue una experiencia grata participar en este proyecto, aprendí de la arquitectura y el manejo de nuevas tecnologías como Jenkins y lenguajes de programación como plsql, aparte de las tecnologías y los lenguajes de programación, fue una gran oportunidad para desarrollar habilidades blandas como la comunicación y el liderazgo en un equipo de trabajo bajo una metodología Scrum, es algo fuera de lo que me imaginaba en un principio de la práctica, pero fue una experiencia gratificante para mí como persona.

9. Referencias Bibliográficas

[1] DevOps. Internet. [https://www.gradiant.org/noticia/devops-negocio-integracion-continua-docker-microservicios/].*José Otero-Pena.13 de Julio de 2017.[Citado el 3 de Marzo de 2018]. Sin restricción de acceso.*

[2] Que es Scrum. Internet. [https://proyectosagiles.org/que-es-scrum/]. Proyectos agiles.org. 2017[Citado el 3 de Marzo de 2018]

[3] Test Driven Developmen.Internet. [https://sg.com.mx/revista/46/test-driven-development]. Alfredo Chavez. 2009.[Citado el 4 de Marzo de 208]

[4] Kanban. Internet. [http://revista.uxnights.com/tag/kanban/]. Victor García. 29 de Agosto de 2016. [Citado el 4 de Marzo de 2018]

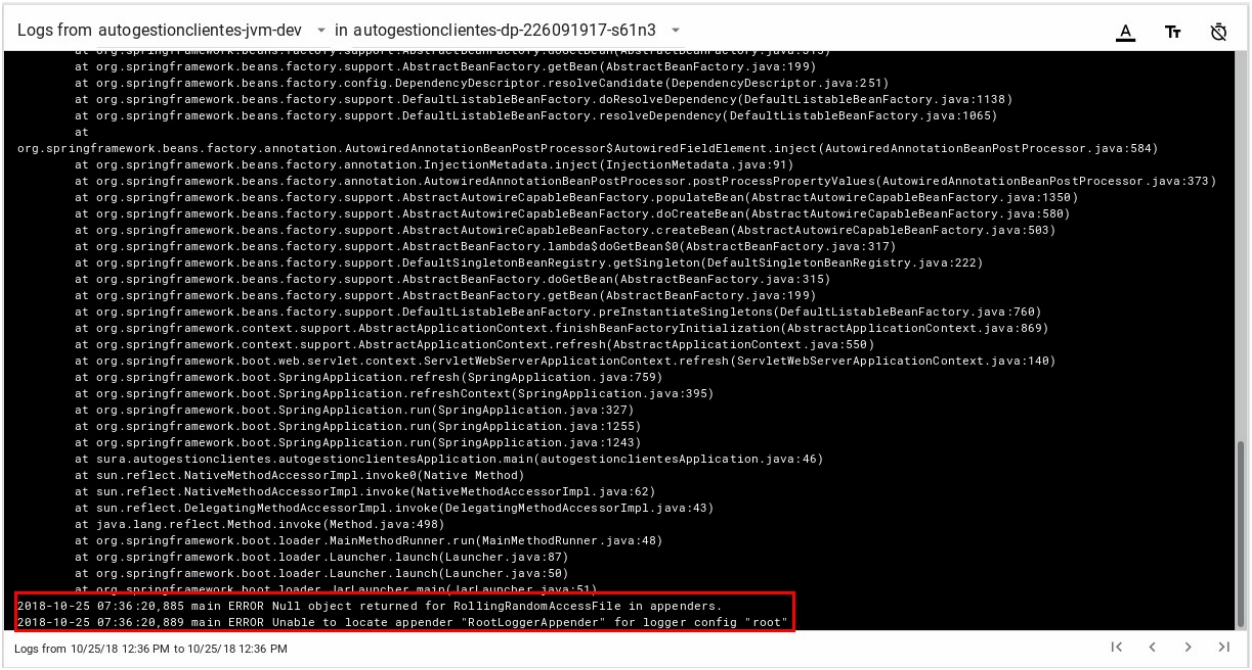
[5] DISEÑO WEB ADAPTATIVO O RESPONSIVO. [http://www.revista.unam.mx/vol.14/num1/art07/art07.pdf]. Esther Labrada Martínez y Cristina Salgado Ceballos. 1 de enero de 2013. [Citado el 5 de Marzo de 2018]

[6] Store Procedure. Internet. [https://searchoracle.techtarget.com/definition/stored-procedure].Margaret Rouse.Febrero de 2017. [Citado el 5 de Diciembre de 2018]

[7] Aplicación web Java autocontenida. Internet. [https://picodotdev.github.io/blog-bitix/2015/03/aplicacion-web-java-autocontenida-con-tomcat-embedded/].14 de Marzo de 2015. [Citado el 5 de Diciembre de 2018]

10. Anexos

Anexo 1. Pantalla principal para la gestión uno a uno.



Anexo 2. Problema presentado a la hora de desplegar en ambiente de desarrollo.