



Universidad Tecnológica Centroamericana

Facultad de Ingeniería

Práctica Profesional

Práctica Profesional, Movitext S.A. de C.V.

Previo a la Obtención del Título

Ingeniería en Sistemas Computacionales

Presentado por:

Diego Ernesto Montes 21441253

Asesor: Ing. Cesar Darío Orellana.

Campus San Pedro Sula

X, 2020

Índice de Contenido

| | |
|-------------------------------------------------|-----------|
| I. Introducción | 6 |
| II. Generalidades de la Empresa | 7 |
| II.1. Descripción de la Empresa | 7 |
| II.2. Descripción del Departamento o Unidad | 7 |
| III. Planteamiento del Problema | 8 |
| III.1. Antecedentes del Problema | 8 |
| III.2. Definición del Problema | 8 |
| III.3. Objetivos | 8 |
| III.3.1. Objetivo General | 8 |
| III.3.2. Objetivos Específicos | 8 |
| IV. Marco Teórico | 10 |
| IV.1. Metodologías de Desarrollo | 10 |
| IV.1.1. Modelo de Prototipos | 10 |
| IV.1.1.1. Recopilación y análisis de requisitos | 11 |
| IV.1.1.2. Diseño rápido | 11 |
| IV.1.1.3. Construcción de prototipos | 11 |
| IV.1.1.4. Evaluación del usuario | 11 |
| IV.1.1.5. Refinamiento | 11 |
| IV.1.1.6. Desarrollo del producto | 12 |
| IV.2. Tecnologías Utilizadas | 12 |
| IV.2.1. MySQL | 12 |
| IV.2.1.1. Bases de datos | 12 |
| IV.2.1.2. SQL | 13 |
| IV.2.1.3. Historia | 14 |
| IV.2.2. PHP | 14 |
| IV.2.2.1. Lenguajes de Scripting | 14 |
| IV.2.2.2. Historia | 15 |
| IV.2.3. Laravel | 15 |
| IV.2.3.1. Modelo-Vista-Controlador | 15 |
| IV.2.3.2. Historia | 16 |
| IV.2.4. Apache | 16 |
| IV.2.4.1. Historia | 17 |
| IV.2.5. Perl | 17 |
| IV.2.5.1. Historia | 17 |
| IV.2.6. Redis | 17 |
| IV.2.6.1. Bases de datos clave-valor | 18 |

| | |
|------------------------------------------------|-----------|
| IV.2.6.2. Historia | 18 |
| IV.2.7. Git | 18 |
| IV.2.7.1. Historia | 19 |
| IV.2.8. Slack | 19 |
| V. Descripción del Trabajo Desarrollado | 20 |
| V.1. Fase de Entrenamiento | 20 |
| V.2. Proyecto de Implementacion de Redis | 24 |
| V.2.1. Instalacion y configuracion | 26 |
| V.2.2. Diseno de estructuras | 27 |
| V.2.3. Implementación | 30 |
| V.3. Redis Monitor | 34 |
| V.3.1. Preparación | 36 |
| V.3.2. Diseño de la aplicación | 37 |
| V.3.3. Desarrollo del proyecto | 38 |
| VI. Conclusiones | 46 |
| VII. Recomendaciones | 47 |
| VIII. Bibliografía | 48 |

Índice de Ilustraciones

| | |
|---------------------------------------------------------------|----|
| Ilustración 1, Diagrama de SQL | 13 |
| Ilustración 2, Diagrama MVC | 16 |
| Ilustración 3, Equipo de trabajo | 20 |
| Ilustración 4, Infraestructura LAMP | 21 |
| Ilustración 5, Formulario de demostración | 23 |
| Ilustración 6, Flujo de mensaje | 24 |
| Ilustración 7, Extensiones del servicio principal | 25 |
| Ilustración 8, Flujo de mensaje con Redis implementado | 26 |
| Ilustración 9, Diseño de estructuras | 28 |
| Ilustración 10, Diseño de estructuras | 28 |
| Ilustración 11, Diseño de estructuras | 29 |
| Ilustración 12, Bitácora de Redis | 32 |
| Ilustración 13, Diapositiva de la presentación del proyecto | 33 |
| Ilustración 14, Diagrama de casos de uso de la aplicación web | 35 |
| Ilustración 15, Demostración de Laravel | 37 |
| Ilustración 16, Diagrama de base de datos | 38 |
| Ilustración 17, Página de aterrizaje | 39 |
| Ilustración 18, Vista de crear estructura | 40 |
| Ilustración 19, Vista de mostrar estructura | 41 |
| Ilustración 20, Vista de creación de estructuras | 42 |
| Ilustración 21, Diagrama de middleware | 43 |

Índice de Tablas

| | |
|------------------------|----|
| Tabla 1, Semanas 1 - 2 | 21 |
| Tabla 2, Semana 3 | 23 |
| Tabla 3, Semana 4 | 27 |
| Tabla 4, Semanas 5 - 6 | 29 |
| Tabla 5, Semana 7 | 30 |
| Tabla 6, Semana 8 | 31 |
| Tabla 7, Semana 9 | 32 |
| Tabla 8, Semana 10 | 33 |
| Tabla 9, Semana 11 | 36 |
| Tabla 10, Semana 12 | 37 |
| Tabla 11, Semana 13 | 39 |
| Tabla 12, Semana 14 | 40 |
| Tabla 13, Semana 15 | 41 |
| Tabla 14, Semana 16 | 43 |
| Tabla 15, Semana 17 | 44 |
| Tabla 16, Semana 18 | 45 |

I. Introducción

En este informe se demuestra el trabajo realizado durante la práctica profesional en la empresa Movitext S.A. de C. V.. En este se describe a detalle todas las actividades que se llevaron a cabo durante la práctica profesional.

En el capítulo II se describe de forma general la empresa Movitext S.A. de C. V., su proceso de negocio, y se brinda una descripción del departamento de IT, el área donde se realizó la práctica.

En el capítulo III se plantea el problema que se estaría solucionando, así como los antecedentes y su definición. En este capítulo se plantean los objetivos generales y los objetivos específicos del proyecto.

En el capítulo IV se presenta el marco teórico del proyecto. En este se desglosan los términos, las tecnologías y las metodologías de desarrollo que se utilizaron al desarrollar el proyecto, entre estas están PHP, Laravel y Redis.

En el capítulo V se describe con detalle el trabajo desarrollado durante la práctica profesional. Este está ordenado cronológicamente, donde cada sección es una semana de trabajo.

II. Generalidades de la Empresa

II.1. Descripción de la Empresa

Movitext es una empresa que provee servicios de mensajería A2P, o aplicación a persona. Movitext proporciona rutas de conexión y marcaciones de operadora para enviar SMS. Esto lo hace a través de una aplicación web donde los clientes pueden administrar su crédito, mandar mensajes en masa, ver reportes, además de otras características que el servicio provee.

II.2. Descripción del Departamento o Unidad

Esta práctica profesional se desarrolló en el departamento de IT. En este departamento se desarrolla y se le da mantenimiento a la aplicación web que manejan los clientes y al servicio principal que se encarga de enrutar y enviar los mensajes de texto.

En este proyecto se tuvo como líder al jefe del departamento de IT. Como miembros del equipo en estos proyectos estaba solamente mi persona. Desde luego se pudo acudir al resto del departamento con cualquier duda durante el desarrollo del proyecto.

III. Planteamiento del Problema

III.1. Antecedentes del Problema

El servicio que se utiliza para hacer el envío y ruteo de los mensajes de texto tiene una característica que nos deja agregar plugins los cuales nos permiten modificar el mensaje antes de que sean enviados. Estos plugins necesitan hacer uso de información que estaba hard coded dentro de cada plugin, lo cual al momento de querer modificar dicha información se requiere hacer un reinicio del servicio principal para que volviera a montar los plugins con los nuevos datos.

La implementación actual de los plugins cumple con las necesidades de los clientes, pero cuando estas necesidades cambian la actualización de datos es tardía y susceptible a errores.

III.2. Definición del Problema

Debido a que la información utilizada por los plugins esta hardcoded dentro de cada plugin, la actualización de estos datos requiere un reinicio total del sistema principal, el cual se realiza en horas de bajo tráfico para minimizar el tráfico perdido. También la actualización de datos es susceptible a errores y si se llegara a actualizar un dato incorrectamente sería necesario un reinicio de emergencia el cual probablemente suceda en horas de alto tráfico.

III.3. Objetivos

III.3.1. Objetivo General

Implementar una base de datos clave-valor en memoria y todo lo relacionado a ella sobre el servicio principal de Movitext para mantener la información disponible a petición.

III.3.2. Objetivos Específicos

1. Familiarizarse con las tecnologías utilizadas en el desarrollo de ambos proyectos.
2. Diseñar las estructuras de datos para almacenar la información que las funcionalidades del servicio principal requieren.

3. Desarrollar nuevas funcionalidades que el servicio principal de Movitext requiere.
4. Desarrollar una aplicación web para poder manipular los datos de configuración de las funcionalidades del sistema principal de Movitext.

IV. Marco Teórico

IV.1. Metodologías de Desarrollo

IV.1.1. Modelo de Prototipos

“Un prototipo es una visión preliminar del modelo futuro, es un modelo operable, fácilmente ampliable y modificable, que tiene todas las características propuestas, pero realmente es un modelo básico que tiene que ser mejorado” (Montemayor Marin, Bijarro Hernandez, Estrada Bellman, 2007).

La “metodología” de prototipo por sí sola no es exactamente una metodología, se puede aplicar a otras metodologías de desarrollo. Usar prototipos reduce el riesgo de un proyecto al dividir el desarrollo en secciones que son más manejables y además involucran al usuario desde el principio. De esta forma el usuario puede proveer retroalimentación sobre cómo espera que el producto se comporte al final. Eso crea una mejora en la adopción al software por parte del usuario. Otra ventaja de los prototipos es que las nuevas ideas se pueden intentar rápidamente sin agregar tanto costo de recursos al proyecto, y al no ser aprobadas o necesitadas se puede desechar con la misma rapidez.

Los prototipos siguen los siguientes 6 pasos:

- Recopilación y análisis de requisitos
- Diseño rápido
- Construcción de prototipos
- Evaluación del usuario
- Refinamiento
- Desarrollo del producto

(Naji, 2016)

IV.1.1.1. Recopilación y análisis de requisitos

En la primera etapa se define de manera detallada todos los requerimientos que el cliente necesita de la aplicación. Esto se logra por medio de una reunión con el cliente para poder llegar a una conclusión, una meta que cumple las necesidades del cliente.

IV.1.1.2. Diseño rápido

Se elabora rápidamente una estructura del prototipo con cada uno de los requerimientos que el cliente solicitó.

IV.1.1.3. Construcción de prototipos

Se comienza a construir el prototipo del software. Este prototipo es solamente una muestra, una demostración, del comportamiento o la visualización del software una vez este ya esté completado. Se basa en el diseño rápido anteriormente definido.

IV.1.1.4. Evaluación del usuario

Como acto siguiente, el usuario comienza a evaluar el prototipo y analiza cada aspecto del mismo. Todo con el fin de identificar las fortalezas y debilidades y así poder comunicar que partes del prototipo se deben mejorar, cambiar o remover completamente.

IV.1.1.5. Refinamiento

El prototipo se modifico tomando en cuenta los comentarios del cliente. Al hacer esto se crea una nueva versión del prototipo, que también será evaluado por el cliente de la misma manera que el prototipo anterior. El fin de esto es ir afinando cada uno de los requisitos del cliente. Este paso se puede repetir cuantas veces sea necesario, hasta que el cliente esté satisfecho.

IV.1.1.6. Desarrollo del producto

Una vez que el cliente está satisfecho con el prototipo, se comienza con el desarrollo oficial del software, lógicamente basándose en el prototipo final, que es el que el cliente aprobó.

IV.2. Tecnologías Utilizadas

IV.2.1. MySQL

MySQL es un sistema de manejo de base de datos de código abierto desarrollado originalmente por una compañía sueca llamada MySQL AB. Al ser una base de datos se utiliza para almacenar y consultar datos al igual que la manipulación de los mismos. Para realizar estas acciones, MySQL utiliza el lenguaje estándar SQL, de ahí el nombre de la aplicación. (MySQL, 2020)

IV.2.1.1. Bases de datos

Una base de datos es una colección organizada de información que persiste sobre un periodo largo de tiempo. El sistema de manejo de base de datos (DBMS, database management system) es el software que interactúa con usuarios finales, aplicaciones, y la base de de datos en sí para capturar y analizar la información. El conjunto de la base de datos, el DBMS, y las aplicaciones asociadas se refieren como el sistema de base de datos. Comúnmente el término base de datos se usa para referirse o a el DBMS, el sistema de base de datos, o a la base de datos en sí.

Un sistema de manejo de base de datos nos permite:

- Crear nuevas bases de datos y especificar la estructura lógica de la información.
- Hacer solicitudes sobre la data guardada en ella.
- Almacenar grandes cantidades de información, a través un largo periodo de tiempo.

- Controlar el acceso a la información por muchos usuarios a la vez, sin permitir que las acciones de un usuario afecten al otro.

(Ullman y Widom, 1997)

IV.2.1.2. SQL

SQL o Structured Query Language (lenguaje de solicitudes estructuradas) es una herramienta utilizada para organizar, manejar, y solicitar información guardada en una base de datos relacional. Aunque su uso más conocido es el de solicitar información, SQL se puede utilizar para manejar todos los aspectos del sistema de manejo de base de datos, incluyendo:

- Definir la estructura y organización de la información almacenada.
- Obtener información almacenada en la base de datos.
- Manipular la información almacenada en la base de datos.
- Controlar el acceso de los usuarios y sus permisos en cuanto a la información almacenada.

(Oppel, Groff, Weinberg, 2010)

A continuación un diagrama de la relación entre SQL, un DBMS y la base de datos:

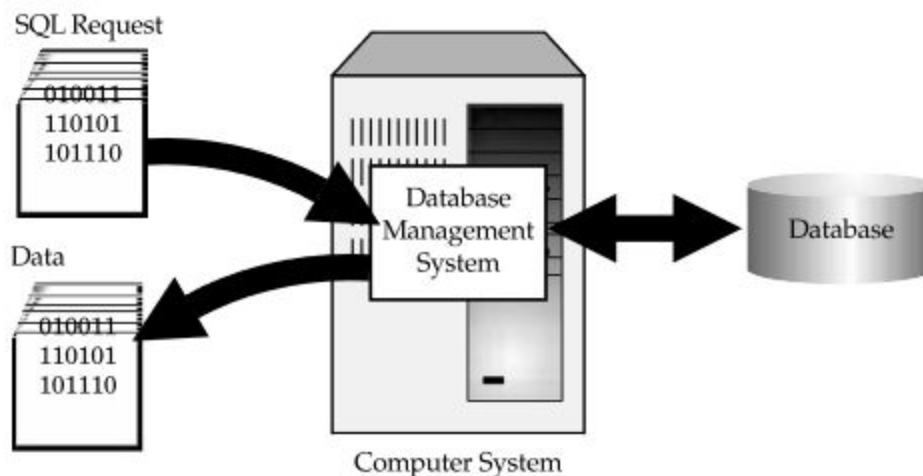


Ilustración 1, Diagrama de SQL

Fuente: (Opper, Groff, Weinberg, 2010)

IV.2.1.3. Historia

Fundada por David Axmark, Allan Larsson y Michael "Monty" Widenius, la compañía sueca MySQL AB lanza la primera versión de MySQL en 1995. Fue creada originalmente para uso personal basado en mSQL. En enero del 2008 Sun Microsystems adquirió MySQL AB y un año más tarde Oracle adquirió Sun Microsystems quienes continúan el desarrollo de MySQL hoy en día. (Monty, 2009)

IV.2.2. PHP

PHP es un lenguaje de scripting de propósito general que es ideal para el desarrollo web ya que puede ser embebido en HTML. Se utiliza para generar páginas web dinámicas, es decir, páginas cuyo contenido no siempre es el mismo. (PHP, 2020)

IV.2.2.1. Lenguajes de Scripting

Un lenguaje de scripting es un lenguaje de programación para un ambiente específico que permite automatizar la ejecución de tareas. Los lenguajes de scripting son comúnmente interpretados, es decir no es compilado previo a ser ejecutado, en su vez existe un proceso al que se le llama interpretador quien se encarga de ejecutar las instrucciones directamente. Las aplicaciones de lenguajes de scripting incluyen aplicaciones de software, páginas web, y terminales de un sistema operativo. (ECMA-262, 2020)

IV.2.2.2. Historia

PHP fue creado originalmente por Rasmus Lerdorf en el año 1995. En 1997 dos programadores israelíes del Technion, Zeev Suraski y Andi Gutmans, reescribieron el analizador sintáctico y crearon la base del PHP3, y cambiaron el significado de las siglas de PHP de Personal Home Page a PHP Hypertext Preprocessor. Hoy en día el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. (PHP, 2020)

IV.2.3. Laravel

Laravel es un marco de trabajo de PHP de código abierto que se utiliza para desarrollar aplicaciones web siguiendo la arquitectura de modelo-vista-controlador. (Laravel, 2020)

IV.2.3.1. Modelo-Vista-Controlador

Modelo-vista-controlador (MVC) es un patrón de arquitectura comúnmente utilizada en aplicaciones web. Este provee tres capas: el modelo, la vista y el controlador. El patrón MVC provee la capacidad de desarrollo en paralelo, ya que cada capa de la aplicación es independiente de las otras, estas se pueden trabajar concurrentemente. Las tres capas se definen a continuación:

- **Modelo.** Las clases modelo se utilizan para implementar la lógica de las estructuras de datos utilizadas por la aplicación. Estas clases se usan para obtener, insertar o actualizar la información en la base de datos asociada a nuestra aplicación.
- **Vista.** Las vistas son las interfaces de la aplicación. Los usuarios finales interactúan con la aplicación a través de las vistas.
- **Controlador.** Las clases controlador se usan para responder a las peticiones del usuario. Estas son el enlace entre el modelo y las vistas, ya que ejecutan los procesos necesarios del modelo y muestran la vista de acuerdo a las peticiones del usuario.

(Freeman, 2013)

A continuación un diagrama de la interacción entre las tres capas de MVC.

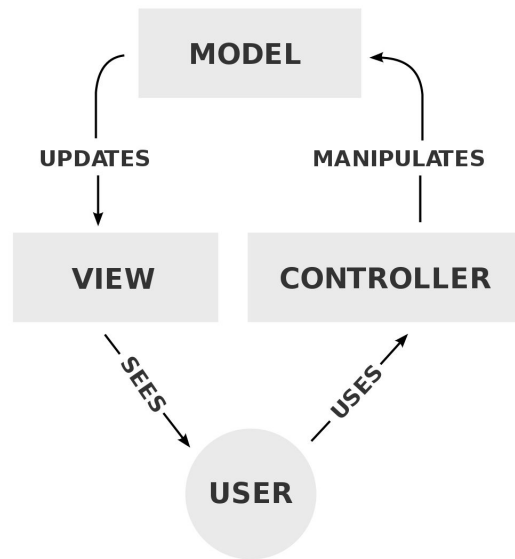


Ilustración 2, Diagrama MVC

Fuente: (RegisFrey, 2010)

IV.2.3.2. Historia

Laravel fue creado por Taylor Otwell para proveer una alternativa al marco de trabajo CodeIgniter, ya que consideraba que ésta carecía de ciertas características que Otwell consideraba esenciales para el desarrollo de aplicaciones web. En junio del año 2011 la primera versión de Laravel es lanzada para proveer las características que a CodeIgniter le faltaban.

(Surguy, 2014)

IV.2.4. Apache

El Servidor HTTP Apache es un software multiplataforma de código abierto que como el nombre sugiere se utiliza para montar servidores web. (Apache, 2020)

IV.2.4.1. Historia

Robert McCool empezó a desarrollar Apache en 1995 basado en el servidor NCSA HTTPd. Apache jugó un papel clave en el desarrollo inicial de la World Wide Web, quien rápidamente desplazó el servidor NCSA HTTPd del mercado. Desde su concepción Apache ha sido desarrollado y mantenido por una comunidad abierta de desarrolladores que hoy trabajan bajo la fundación Apache Software Foundation. (Apache, 2020)

IV.2.5. Perl

Perl es un lenguaje de programación dinámico de alto nivel, interpretado y de propósito general. Fue creado originalmente para hacer más fácil el procesamiento de reportes, por eso le han dado el retroacrónimo "Practical Extraction and Reporting Language". (Perl, 2020)

IV.2.5.1. Historia

En 1987 Larry Wall empezó con el desarrollo de Perl mientras laboraba en Unisys, y en diciembre del mismo año lanzó la primera versión. A través de los próximos años el lenguaje se expandió rápidamente, lanzando 3 versiones nuevas en menos de 5 años. (Wall, Orwant, Christiansen, 2000)

IV.2.6. Redis

Redis es un software libre (licencia BSD) que actúa como una tienda de estructuras de datos en memoria y se considera una base de datos clave-valor. Se utiliza como caché para

guardar datos que tienen que estar fácilmente disponibles. Redis soporta varias estructuras de datos incluyendo hashes, listas, mapas, sets, sorted sets, mapas de bits, entre otros. (Redis, 2020)

IV.2.6.1. Bases de datos clave-valor

Una base de datos clave-valor es un tipo de base de datos no relacional que usa un simple método de clave-valor para almacenar información. Una base de datos clave-valor almacena información en pares de clave-valor en la cual la clave es un identificador único. A diferencia de las bases de datos relacionales, las cuales definen previamente la estructura de la información como una serie de tablas que contiene campos con tipos de datos específicos, las bases de datos clave-valor manejan la información como una sola colección de datos. (Deka, 2017)

IV.2.6.2. Historia

El proyecto de Redis comenzó cuando Salvatore Sanfilippo buscaba formas de mejorar la escalabilidad de su empresa emergente. Al encontrar problemas en la escalabilidad de ciertas cargas de trabajo utilizando bases de datos tradicionales, Sanfilippo prototipo la primera versión de Redis en el lenguaje de programación Tcl. Luego Sanfilippo tradujo el prototipo a C y después de unas semanas de éxito utilizando Redis internamente lanzó la primera versión de Redis en mayo del 2009. (Redis, 2020)

IV.2.7. Git

Git y los programas de control de versiones son herramientas que registran los cambios realizados a un archivo o a varios archivos. Esto permite tener un control sobre las diferentes versiones de un archivo o proyecto. (Git, 2020)

IV.2.7.1. Historia

El desarrollo de Git se dio en 2005 cuando la comunidad que desarrolló el kernel de Linux paro de utilizar el sistema BitKeeper para llevar control sobre el desarrollo del proyecto. A raíz de esto, la comunidad de Linux comenzó a desarrollar su propia herramienta para llevar el control de versiones. (Git, 2020)

IV.2.8. Slack

Slack es una herramienta de colaboración y comunicación para los miembros de una organización. Slack está creado para ayudar a llevar a cabo proyectos. Slack permite la comunicación escrita, llamadas de audio y de video. También permite compartir documentos y compartir pantallas de escritorio. (Slack, 2020)

V. Descripción del Trabajo Desarrollado

En esta sección se describe el trabajo realizado en la empresa Movitext S.A de C.V.. Se me asignó el puesto de desarrollador. Me asignaron dos proyectos a lo largo de mi práctica: la implementación de la nueva herramienta Redis y una aplicación web para monitorear este nuevo servicio. A continuación se detallaran las actividades realizadas desde la inducción a la empresa hasta la realización de ambos proyectos. Los días de trabajo fueron de lunes a viernes, con un horario de 8am a 5pm.

La duración de esta practica profesional en la empresa Movitext S.A. de C.V. fue de de 20 semanas, iniciando el 23 de septiembre y terminando 10 de febrero.

V.1. Fase de Entrenamiento

En las primeras semanas se tuvo reunión con el resto del equipo y con mi jefe directo sobre cuales serian mis responsabilidades y se dedicó tiempo para la capacitación sobre los diferentes servicios existentes en Movitext S.A. de C.V. y su interconexión. También se definieron los temas de los cuales se debía capacitar y se llevó acabo el aprendizaje a través de video tutoriales.

A continuación, un diagrama del equipo de trabajo:

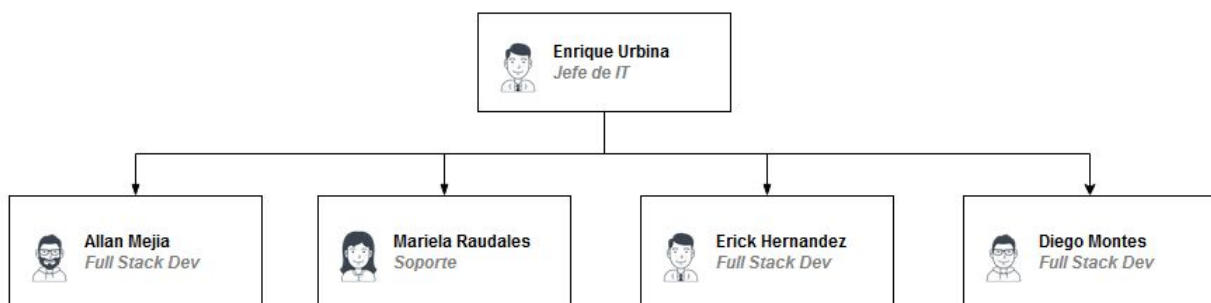


Ilustración 3, Equipo de trabajo

Fuente: Propia, 2020

El entrenamiento se realizó sobre los lenguajes Perl y PHP, que son los lenguajes usados en las tecnologías manejadas en la empresa. Al no tener previa experiencia en ninguno de los lenguajes anteriores se enfoca en profundizar en ambos temas.

Logotipos de las tecnologías aprendidas:



Ilustración 4, Infraestructura LAMP

Fuente: (Falconinternet, 2020)

El ambiente de desarrollo web que se utilizó fue el de la infraestructura de internet LAMP que se refiere a las cuatro tecnologías: Linux como sistema operativo, Apache como servidor de http, MySQL como base de datos y PHP como el lenguaje de programación. Todo esto montado sobre un subsistema Linux dentro de Windows 10.

Tabla 1, Semanas 1 - 2

| Semana | Descripción | Logros |
|----------|---------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| Semana 1 | <ul style="list-style-type: none">Se llevó a cabo una reunión con el equipo para definir puntualmente | <ul style="list-style-type: none">Se logró familiarizarse con el equipo y el |

| | | |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>las responsabilidades asignadas para la práctica profesional.</p> <ul style="list-style-type: none"> ● Se llevó a cabo una reunión con un compañero de equipo con el objetivo de aprender sobre los diferentes sistemas de la empresa. ● Se comenzó el entrenamiento sobre la primera herramienta: PHP. | <p>ambiente de trabajo.</p> <ul style="list-style-type: none"> ● Se aprendió sobre los diferentes sistemas, herramientas y metodologías utilizadas. ● Se estableció el ambiente de desarrollo local de PHP. |
| Semana 2 | <ul style="list-style-type: none"> ● Se reanudó el curso sobre PHP. ● Se realizó el curso sobre la segunda herramienta: Perl. | <ul style="list-style-type: none"> ● Se realizó el desarrollo de una demostración de las habilidades adquiridas. ● Se estableció el ambiente de desarrollo de Perl y se desarrolló el curso. |

Fuente: Propia, 2020

Debido a que PHP es un lenguaje embebido en HTML al programar en este lenguaje se pusieron en práctica los conocimientos de HTML.

A continuación una imagen del formulario desarrollado como demostración del aprendizaje en PHP:

Register

* required field.

* You must agree to terms

Name: *

E-mail: *

Time:

Classes:

Gender: Female Male *

Select:

Agree

Ilustración 5, Formulario de demostración

Fuente: Propia, 2020

Una vez terminado el curso de Perl se tuvo una reunión con mi jefe directo para discutir el primer proyecto al que estaría asignado. En esta reunión se definió el alcance y los requerimientos del proyecto.

Tabla 2, Semana 3

| Semana | Descripción | Logros |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 3 | <ul style="list-style-type: none">Se finalizó el curso sobre Perl.Se llevó a cabo una reunión con el jefe para definir el primer proyecto e introducir la tercera herramienta: Redis.Se aprendió sobre la funcionalidad de Redis y sus características. | <ul style="list-style-type: none">Se realizó el desarrollo de un script que maneja archivos para demostrar las habilidades adquiridas en Perl.Se definieron los requerimientos y la finalidad del primer proyecto.Se familiarizo con la |

| | | |
|--|--|-------------------------------------------------------------|
| | | documentación de Redis, su funcionalidad y características. |
|--|--|-------------------------------------------------------------|

V.2. Proyecto de Implementacion de Redis

El servicio principal de Movitext S.A. de C.V. utiliza plugins en el lenguaje Perl para hacer modificaciones a los mensajes en base a cierta información antes de que estos sean enviados.

Ejemplo de un plugin que rechaza mensajes en base a su contenido:

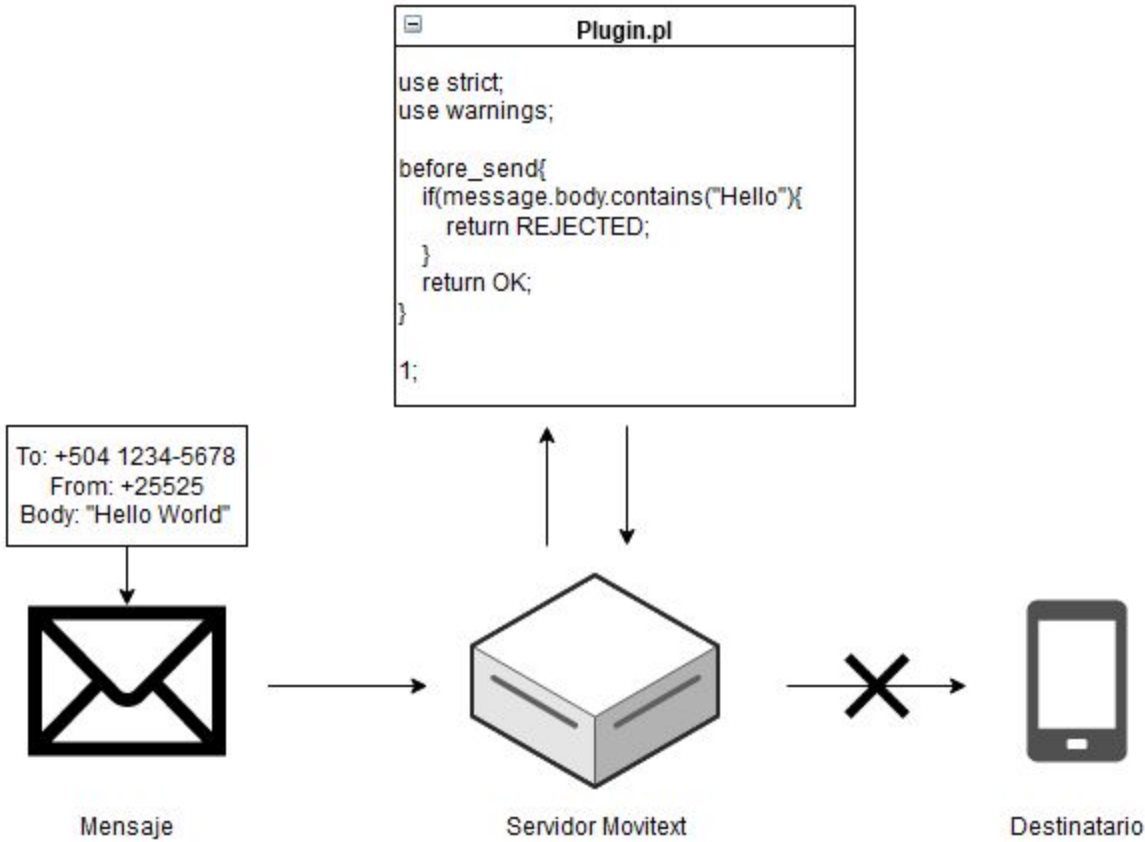


Ilustración 6, Flujo de mensaje

Fuente: Propia 2020

Como se puede ver en la ilustración previa los plugins pueden ser usados para rechazar mensajes basados en ciertas propiedades, ya sea marcación o contenido, o para enrutar los mensajes basados en cierta información. Cuando se comenzó la práctica profesional esta información que define los cambios a los mensajes por enviar estaba guardada en disco. Estos archivos se levantaban únicamente cuando el mismo servicio se levantaba, por lo tanto era imposible modificar esta información en caliente.

En el siguiente diagrama se puede ver el flujo del procesamiento de un mensaje en el servicio principal y el rol de los plugins en la modificación de estos.

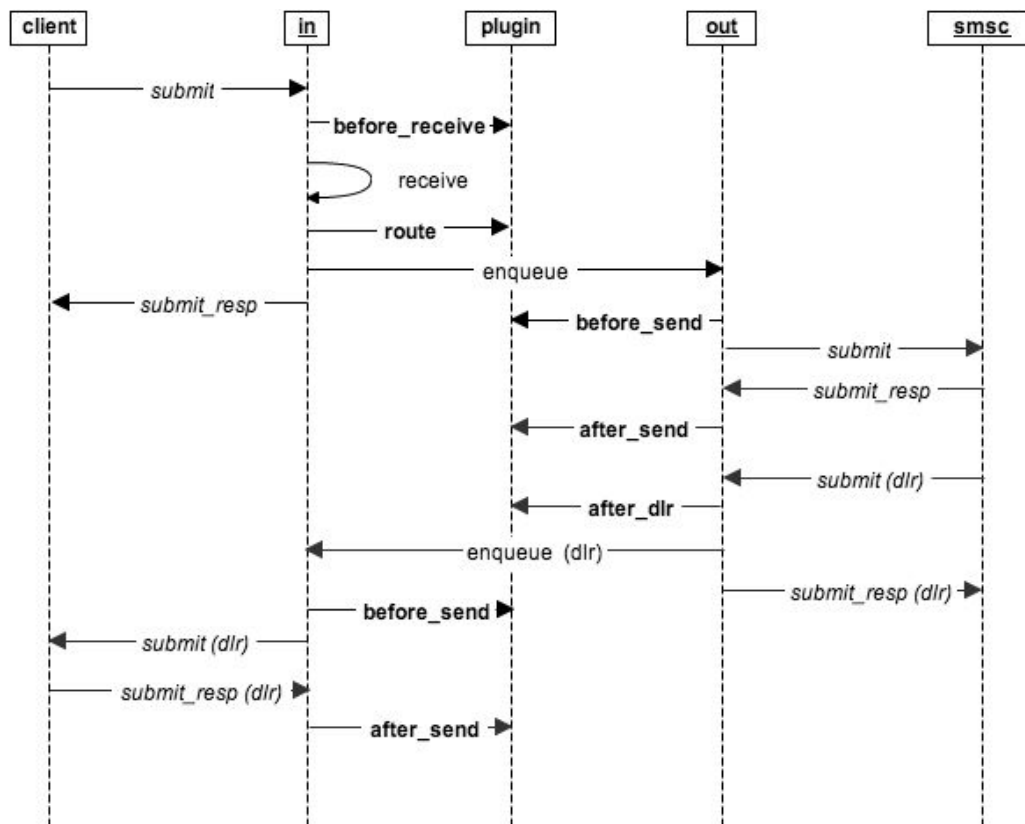


Ilustración 7, Extensiones del servicio principal

Fuente: (EMG, 2016)

El proyecto de implementación de Redis busca mover esta información a un nuevo servicio, donde podrá ser consultada a petición por los plugins y así mismo los datos podrán ser modificados en cualquier momento. Así cuando un cliente solicite un cambio a la información se podrá cumplir la solicitud en tiempo real y no esperar una hora con bajo movimiento para hacer un reinicio total del sistema con la información actualizada.

Ejemplo de rechazo de mensaje en base a su contenido con Redis implementado:

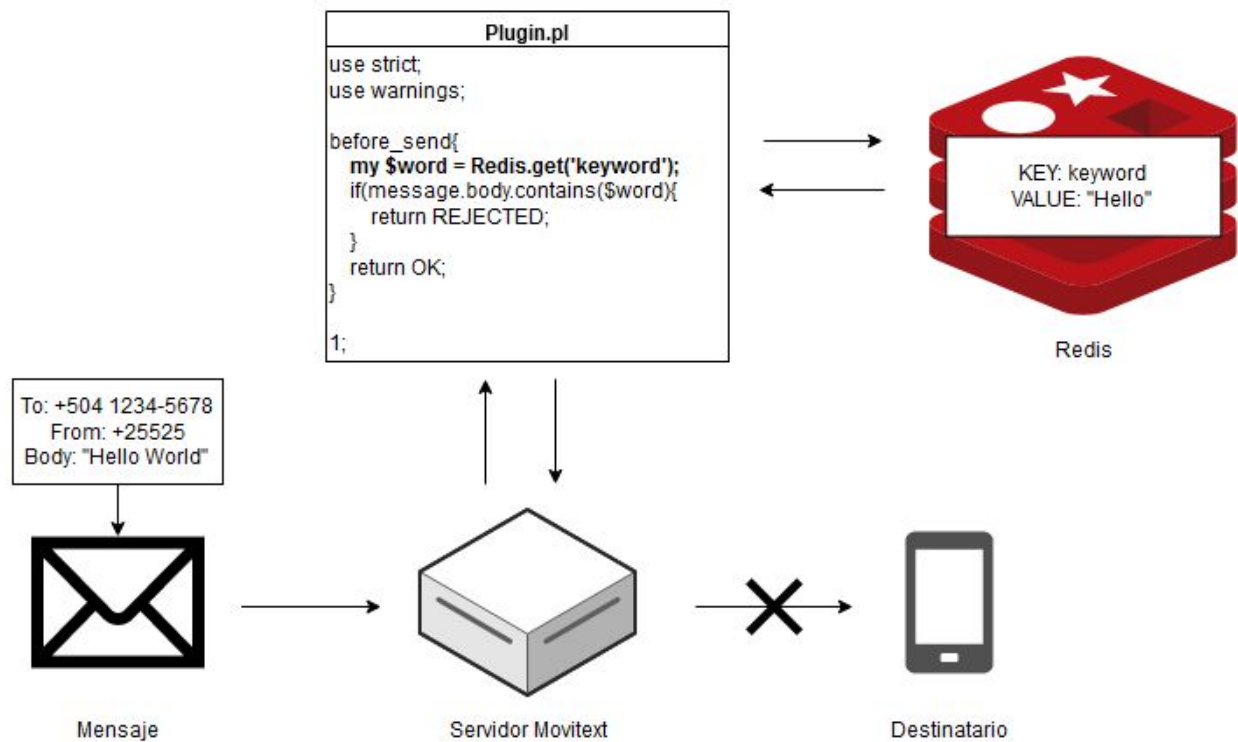


Ilustración 8, Flujo de mensaje con Redis implementado

Fuente: Propia, 2020

V.2.1. Instalacion y configuracion

La instalacion y configuracion de Redis se llevo a cabo de acuerdo a las necesidades del proyecto. Se buscó una versión que fuera compatible a través de los diferentes sistemas operativos implementados en la empresa para que cualquier sistema tenga acceso directo a la

información que este guarda. La instalación en el ambiente local se llevó a cabo en paralelo a la instalación en el ambiente de pruebas para asegurar la compatibilidad de todos los componentes.

El daemon script de Redis por defecto tenía unas incompatibilidades con la versión de Linux del ambiente de desarrollo que se tuvieron que arreglar.

Se instalaron también las librerías necesarias a través del manejador de paquetes de Perl, CPAN y a continuación se comprobó el funcionamiento.

Tabla 3, Semana 4

| Semana | Descripción | Logros |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 4 | <ul style="list-style-type: none"> • Se realizó la instalación y configuración de Redis. • Se aseguró del funcionamiento y que la interacción entre Perl y Redis fuera fiable. | <ul style="list-style-type: none"> • Se logró la instalación exitosa de Redis en los ambientes locales y de pruebas. • Se desarrollaron pequeños scripts para demostrar el funcionamiento de Redis en Perl. |

Fuente: Propia, 2020

V.2.2. Diseño de estructuras

Previamente la información utilizada por los plugins estaba guardada en archivos de formato csv o hardcoded en el plugin en sí. Se analizaron todos los plugins y se recopiló toda la información de la que estos hacen uso. Se transformó esta información a las estructuras de datos compatibles en Redis. Las estructuras de Redis utilizadas fueron Hashes y Sorted Sets.

La información que estaba hardcoded en los plugins utilizaban el tipo de variable hash nativo de Perl, la cual fácilmente se transformó a Hashes de Redis.

A continuación un ejemplo de como se definió el formato de los Hashes para la información hardcoded:

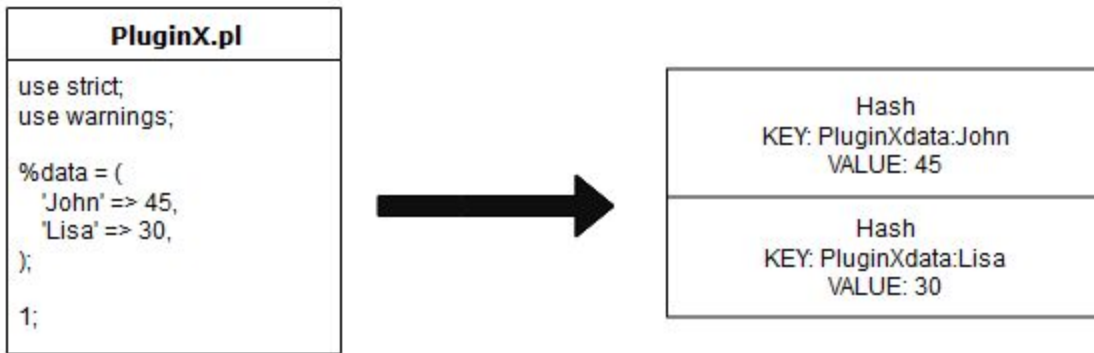


Ilustración 9, Diseño de estructuras

Fuente: Propia, 2020

Parte de la información guardada en formato csv se recorría linealmente en los plugins para buscar una igualdad con los primeros campos y se devuelve el último. Estos archivos se transformaron en estructuras Hash dentro de Redis convirtiendo los primeros campos en parte de la llave y el último campo en el valor que la llave contenía.

A continuación un ejemplo de como se definió el formato de los Hashes para ciertos archivos en formato csv:

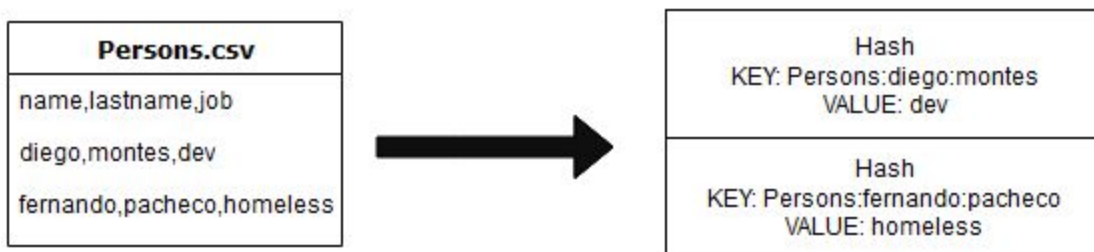


Ilustración 10, Diseño de estructuras

Fuente: Propia, 2020

Se desarrolló un script que se utiliza para subir toda la información inicialmente a Redis en base a archivos en formato csv. Ya que el script trabaja con archivos csv, no fue necesario modificar los archivos que previamente guardaban la información.

Tabla 4, Semanas 5 - 6

| Semana | Descripción | Logros |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 5 | <ul style="list-style-type: none"> Se analizaron los plugins a los cuales se les implementa la conexión Redis. Se inició el diseño de las nuevas estructuras de datos. | <ul style="list-style-type: none"> Se recopiló toda la información que será movida a Redis. Se transformó parte de la información recopilada a estructuras de datos que Redis soporta. |
| Semana 6 | <ul style="list-style-type: none"> Se finalizó el diseño de las estructuras de datos. Se monto la informacion en Redis. | <ul style="list-style-type: none"> Se transformó el resto de la información recopilada a estructuras de datos que Redis soporta. Se desarrolló un script para subir automáticamente toda la información transformada. |

Fuente: Propia, 2020

Varios archivos en formato csv guardan rangos de números y cada rango tiene un valor correspondiente. Los plugins que los utilizan los recorren linealmente para encontrar el rango que al que pertenece cierto número y conseguir el valor de ese rango. Estos rangos se transformaron en Sorted Sets y el valor correspondiente de cada rango se guardó en un Hash.

A continuación un ejemplo de cómo se transformaron los rangos a estructuras de Redis:

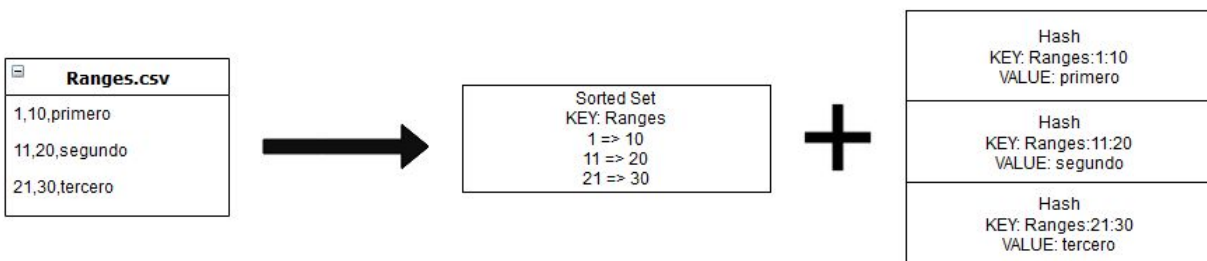


Ilustración 11, Diseño de estructuras

Fuente: Propia, 2020

Para el manejo de estas estructuras híbridas se creó un módulo dentro de Redis que recibe el número a buscar y la estructura donde lo buscare, y regresa el valor correspondiente al rango que el número pertenece dentro de la estructura especificada. El módulo se desarrolló en C y se implementó una búsqueda binaria para agilizar el procesamiento de esta consulta.

Tabla 5, Semana 7

| Semana | Descripción | Logros |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 7 | <ul style="list-style-type: none"> • Se identificó la necesidad de un módulo para manejar las estructuras híbridas. • Se implementó una búsqueda binaria sobre los rangos. | <ul style="list-style-type: none"> • Se desarrolló un módulo para Redis que implementa una búsqueda binaria sobre un Sorted Set, y devuelve el valor del Hash que le corresponde. |

Fuente: Propia, 2020

V.2.3. Implementación

Luego de haber montado toda la información en Redis se reescribieron todos los plugins que harán uso de él. Se creó un nuevo proyecto en BitBucket donde se guardarán todos los plugins ya que no existía uno previamente. Los plugins se reescribieron desde cero ya que contenían lógica hardcoded, código inalcanzable y código que ya no se utiliza debido a la implementación de Redis.

Se crearon scripts para generar datos al azar para hacer pruebas con los plugins. Todos los plugins se desarrollaron con pruebas unitarias asegurándose de que el comportamiento del nuevo plugin sea exactamente igual al de la implementación actual.

Durante las pruebas con datos al azar se presentó un problema en los plugins que utilizan el módulo de búsqueda de rangos, este no devuelve correctamente los datos. Al investigar y

realizar pruebas al módulo de rangos se identificó una falla en la lógica de la implementación de la búsqueda binaria la cual se arregló de inmediato.

Tabla 6, Semana 8

| Semana | Descripción | Logros |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 8 | <ul style="list-style-type: none"> ● Se creó un nuevo proyecto de git. ● Se desarrollaron los nuevos plugins. ● Se identificó un problema con el módulo de búsqueda de rangos. | <ul style="list-style-type: none"> ● Se creó un nuevo proyecto en BitBucket donde se llevará control de las versiones de los plugins. ● Se realizó el desarrollo de los plugins utilizando pruebas unitarias y pruebas al azar. ● Se corrigió el problema que presento el modulo de busqueda de rangos. |

Fuente: Propia, 2020

Luego de completar el desarrollo de los plugins se montó toda la información al ambiente de pruebas y se implementaron los nuevos plugins. Con la ayuda de un compañero de equipo se realizaron pruebas individuales a todos los plugins, cambiando la información en Redis y revisando el comportamiento de los plugins a través de las bitácoras.

Durante estas pruebas se presentó otro error con el módulo de búsqueda en rangos que hacía que Redis se detuviera. A continuación parte de la bitácora de Redis durante este error:

```

----- STACK TRACE -----
EIP:
/usr/local/bin/redis-server 127.0.0.1:6379(addReplyBulkLen+0x0)[0x7f47966441f0]

Backtrace:
/usr/local/bin/redis-server 127.0.0.1:6379(logStackTrace+0x5a)[0x7f47966764ea]
/usr/local/bin/redis-server 127.0.0.1:6379(sigsegvHandler+0xb1)[0x7f4796676ca1]
/lib/x86_64-linux-gnu/libpthread.so.0(+0x12890)[0x7f4795a42890]
/usr/local/bin/redis-server 127.0.0.1:6379(addReplyBulkLen+0x0)[0x7f47966441f0]
/usr/local/bin/redis-server 127.0.0.1:6379(addReplyBulk+0x11)[0x7f4796644331]
/usr/local/bin/redis-server 127.0.0.1:6379(RM_ReplyWithString+0x1b)[0x7f479669fecb]
/home/diegu7-mt/diegu7/redis/searchcarrier.so(SearchCarrier_RedisCommand+0x535)[0x7f47935c44b2]
/usr/local/bin/redis-server 127.0.0.1:6379(RedisModuleCommandDispatcher+0x62)[0x7f47966a13f2]
/usr/local/bin/redis-server 127.0.0.1:6379(call+0xab)[0x7f47966348bb]
/usr/local/bin/redis-server 127.0.0.1:6379(processCommand+0x42f)[0x7f479663507f]
/usr/local/bin/redis-server 127.0.0.1:6379(processInputBuffer+0x115)[0x7f4796645115]
/usr/local/bin/redis-server 127.0.0.1:6379(aeProcessEvents+0x148)[0x7f479662e638]
/usr/local/bin/redis-server 127.0.0.1:6379(aeMain+0x2b)[0x7f479662ea1b]
/usr/local/bin/redis-server 127.0.0.1:6379(main+0x4ec)[0x7f479662b54c]
/lib/x86_64-linux-gnu/libc.so.6(__libc_start_main+0xe7)[0x7f4795651b97]
/usr/local/bin/redis-server 127.0.0.1:6379(_start+0x2a)[0x7f479662b83a]

```

Ilustración 12, Bitácora de Redis

Fuente: Propia, 2020

Al investigar el problema en la bitacora de Redis y tratar de recrear el problema se identificó que uno de los rangos contenía un espacio y al convertirlo a un número usable, el modulo de busqueda detenía fatalmente a Redis. Se corrigió el dato y se reanudaron las pruebas.

Tabla 7, Semana 9

| Semana | Descripción | Logros |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 9 | <ul style="list-style-type: none"> Se realizaron pruebas a los nuevos plugins. Se identificó un problema con el módulo de búsqueda de rangos. | <ul style="list-style-type: none"> Se realizaron pruebas puntuales a cada uno de los nuevos plugins y se examinó su comportamiento de cerca. Se corrigieron los datos que provocaron el detenimiento fatal de Redis. |

Fuente: Propia, 2020

Una vez se comprobó el funcionamiento de los plugins en el ambiente de pruebas se llevó a cabo la instalación y configuracion de Redis en el ambiente de producción. Asi como la

instalacion de la libreria de Redis para Perl utilizando CPAN. Ya con Redis instalado se procedió a subir los datos a Redis y posteriormente se implementaron los nuevos plugins al servicio principal.

Una vez terminada la implementación se redactó un documento altamente detallado que describe la implementacion de Redis en el ambiente de producción, desde la instalación y configuración, el desglose de cada una de las estructuras presentes en Redis, una breve descripcion de los plugins y cómo hacen uso de las estructuras en Redis, hasta como interactuar con Redis a traves de la terminal para hacer consultas y cambios.

Para acompañar la documentación se realizó una presentación ante todo el equipo. A continuación una de las diapositivas donde se explican los comandos de Redis:



Comandos

General:

INFO [SECTION]
AUTH PASSWORD
KEYS PATTERN
DEL KEY

Sorted Sets:

ZADD KEY SCORE1 VALUE1.. SCOREN VALUEN
ZRANGE KEY START STOP [WITHSCORES]
ZREVRANGE KEY START STOP [WITHSCORES]
ZSCORE KEY MEMBER

Hashes:

HSET KEY_NAME FIELD VALUE
HGET KEY_NAME FIELD

Ilustración 13, Diapositiva de la presentación del proyecto

Fuente: Propia, 2020

En la presentación se expone lo contenido en la documentación, además de casos de uso puntuales y con una sesión de preguntas y respuestas para acabar con las dudas sobre la nueva herramienta implementada.

Tabla 8, Semana 10

| Semana | Descripción | Logros |
|--------|-------------|--------|
|--------|-------------|--------|

| | | |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 10 | <ul style="list-style-type: none"> ● Se implementaron los nuevos plugins en el ambiente de producción. ● Se documentó el trabajo realizado. ● Se presentó el proyecto ante el equipo. | <ul style="list-style-type: none"> ● Se logró la implementación de los nuevos plugins en el ambiente de producción con éxito. ● Se redactó un documento detallado sobre el trabajo realizado y que describe su utilidad. ● Se realizó una presentación ante el equipo para capacitarlos en el uso de la nueva herramienta. |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Fuente: Propia, 2020

V.3. Redis Monitor

El segundo proyecto, la aplicación web para monitorear Redis, busca dar acceso fácil y amigable a la información que está guardada en Redis, así como facilitar la actualización de esta. También se busca poder monitorear el estado y las bitácoras del servicio de Redis como tal. La implementación de esta aplicación mejorará la interacción entre el equipo de trabajo y la información guardada en Redis a través de una interfaz visual.

A continuación un diagrama de casos de uso:

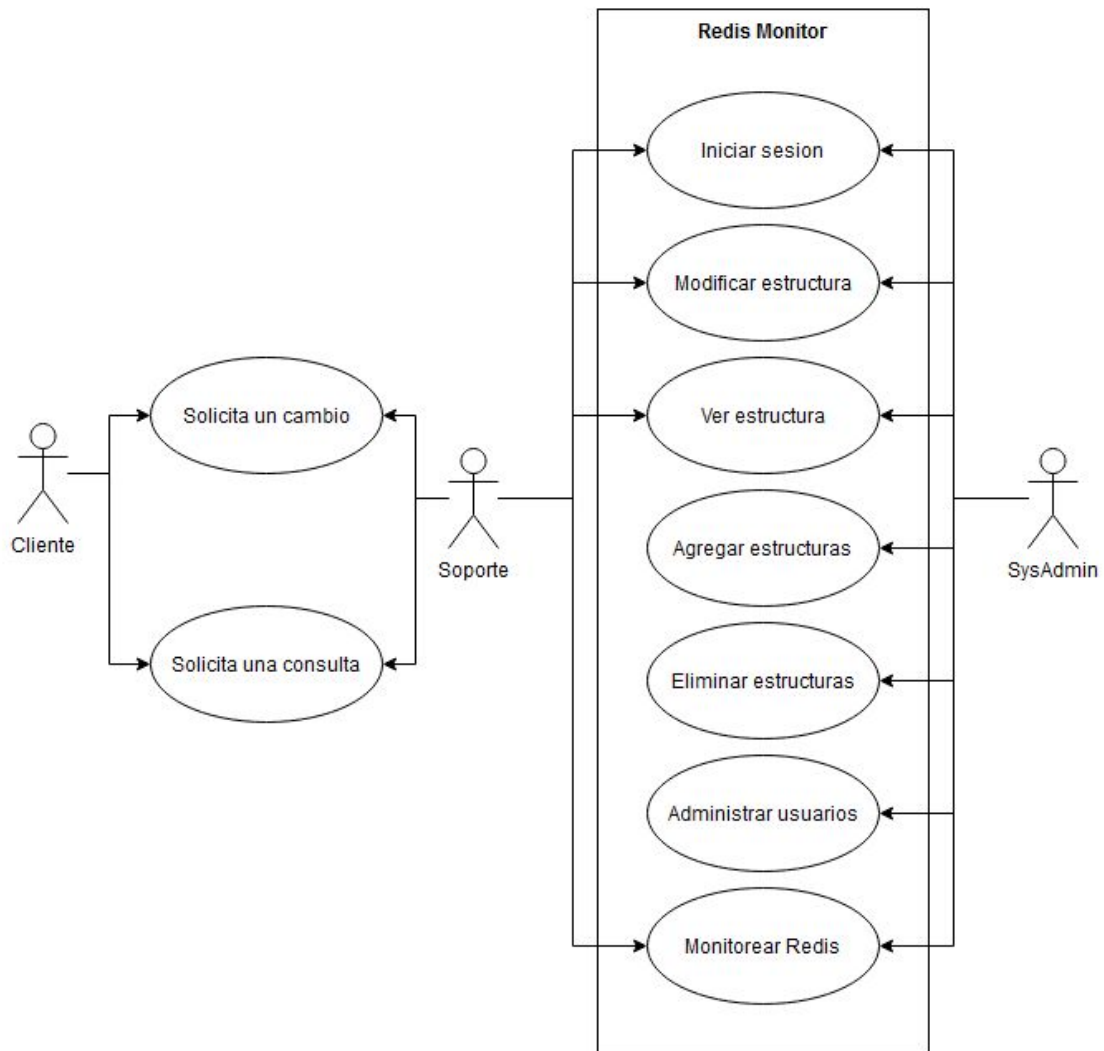


Ilustración 14, Diagrama de casos de uso de la aplicación web

Fuente: Propia, 2020

Este proyecto se dio a luz durante el desarrollo del proyecto anterior. Se vio la necesidad de tener una herramienta que nos permita visualizar todas las estructuras almacenadas en Redis así como los datos que contienen y una descripción del uso que se les da.

La aplicación será desarrollada en la infraestructura de internet LAMP que se refiere a las cuatro tecnologías: Linux como sistema operativo, Apache como servidor de http, MySQL como

base de datos y PHP como el lenguaje de programación. Y se utilizara el marco de trabajo de PHP llamado Laravel que sigue la arquitectura MVC o modelo-vista-controlador.

V.3.1. Preparación

Al finalizar el proyecto anterior se llevó a cabo una reunión con mi jefe directo en el cual se discutió el alcance del proyecto así como los requerimientos y tecnologías a usar. Ya que se desconoce el marco de trabajo elegido para la implementación se llevó a cabo un curso de capacitación para dicha herramienta, en la cual se desarrolló un pequeño proyecto de práctica.

Tabla 9, Semana 11

| Semana | Descripción | Logros |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 11 | <ul style="list-style-type: none"> ● Se llevó a cabo una reunión con el jefe para definir el segundo proyecto. ● Se realizó un curso sobre el marco de trabajo Laravel. | <ul style="list-style-type: none"> ● Se definieron los requerimientos y la finalidad del segundo proyecto. ● Se estableció el ambiente de desarrollo local de PHP con Laravel. ● Se realizó el desarrollo de una demostración de las habilidades adquiridas. |

Fuente: Propia, 2020

Dado que se tenía previo conocimiento sobre la arquitectura MVC no se me dificultó aprender este nuevo marco de trabajo. A continuación una pequeña captura de la aplicación que se desarrolló durante el curso:

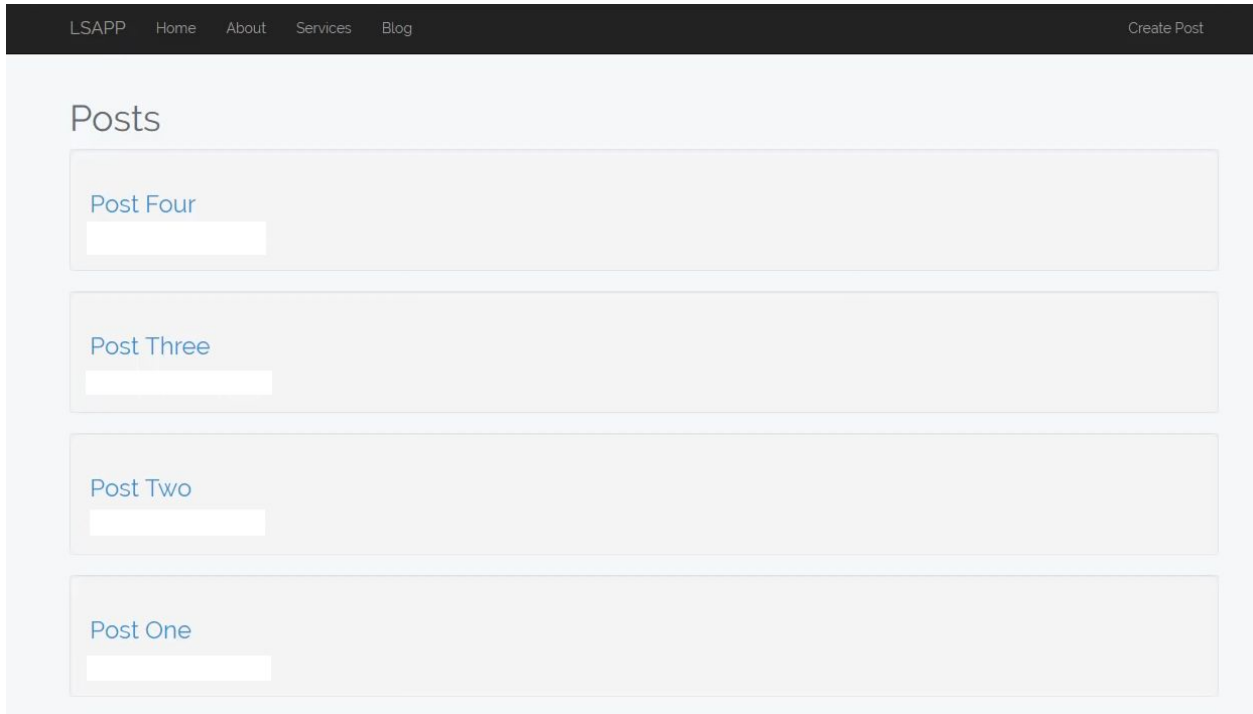


Ilustración 15, Demostración de Laravel

Fuente: Propia, 2020

V.3.2. Diseño de la aplicación

En la siguiente semana se llevó a cabo el diseño de la base de datos de acuerdo a los requerimientos. También se creó un nuevo repositorio donde iría el código fuente de la aplicación y se creó el nuevo proyecto en Laravel.

Tabla 10, Semana 12

| Semana | Descripción | Logros |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 12 | <ul style="list-style-type: none"> • Se creó un nuevo proyecto de git. • Se diseñó la base de datos. • Se creó la base de datos. | <ul style="list-style-type: none"> • Se creó un nuevo proyecto en BitBucket donde se llevará control de las versiones de la aplicación. • Se hizo el diseño inicial de la base de datos del proyecto. • Con el diseño |

| | | |
|--|--|----------------------------------------------------|
| | | propuesto se creó la base de datos en el proyecto. |
|--|--|----------------------------------------------------|

Fuente: Propia, 2020

Utilizando las migraciones de Laravel se creó la base de datos inicial. Laravel también nos permite crear semillas para la base de datos, las cuales serán muy útiles para poder prever cómo se miraría la aplicación con información real. A continuación el diagrama de la base de datos inicial:

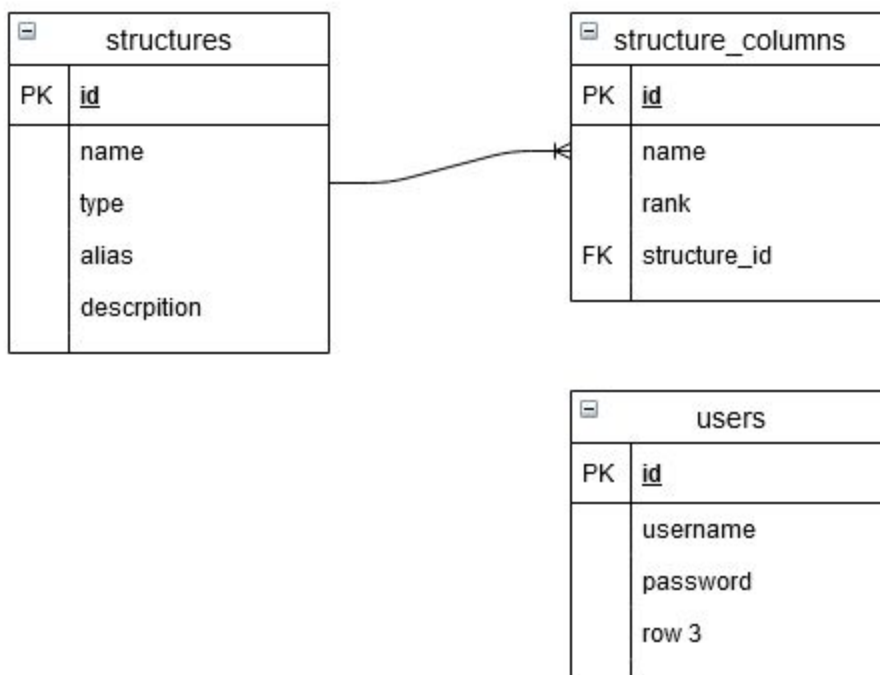


Ilustración 16, Diagrama de base de datos

Fuente: Propia, 2020

V.3.3. Desarrollo del proyecto

Durante las próximas semanas se llevó a cabo el desarrollo de la aplicación. Se comenzó programando un diseño básico de la aplicación donde se agrega una barra de navegación. Se

desarrolló un controlador de páginas para manejar la navegación en la aplicación a través de las páginas que mostrarán información, empezando con la página de aterrizaje, que por el momento regresa un simple view con el título de la aplicación como se puede visualizar a continuación:

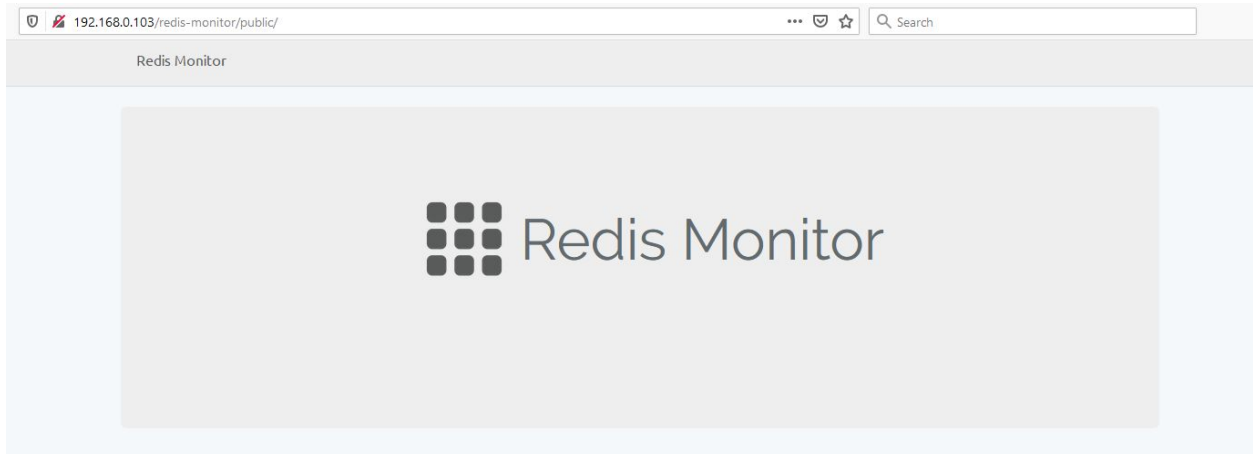


Ilustración 17, Página de aterrizaje

Fuente: Propia, 2020

Se creó también un controlador para las estructuras de datos y se programaron los CRUD así como los endpoints a estos. Se creó también la vista donde se estarían mostrando todas las estructuras disponibles.

Tabla 11, Semana 13

| Semana | Descripción | Logros |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 13 | <ul style="list-style-type: none"> • Se comenzó el desarrollo del front-end. • Se desarrollaron los CRUD de las estructuras de datos. | <ul style="list-style-type: none"> • Se desarrolló una página de aterrizaje básica. • Se programó los CRUD de las estructuras de datos así como la vista que despliega las estructuras disponibles. |

Fuente: Propia, 2020

Luego se realizó el desarrollo de las vistas restantes para las estructuras. Se agregó un botón para agregar estructuras el que lleva a la nueva vista que nos permite agregar estructuras. Por el momento se considera solo un tipo de estructuras que son los tipo hash, las cuales pueden tener n cantidad de columnas. En este punto del proyecto se desafió mi conocimiento de Javascript ya que se agregó un campo de texto dinámico en el que podía agregar la cantidad de columnas necesarias, como se puede observar en la siguiente imagen:

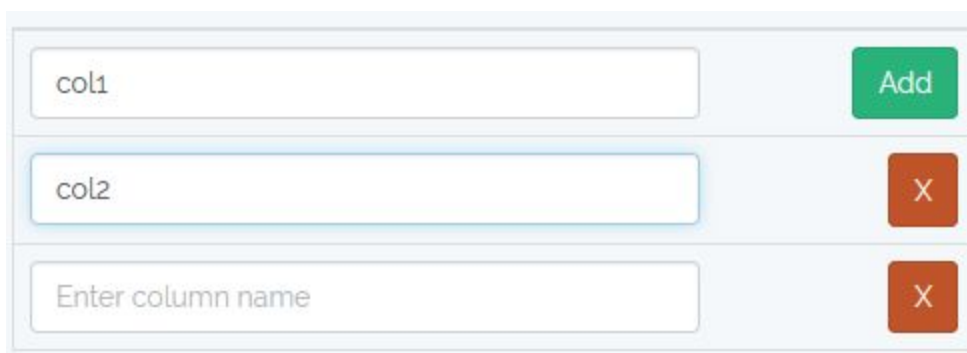


Ilustración 18, Vista de crear estructura

Fuente: Propia, 2020

La vista de editar estructuras se creó en paralelo ya que comparte casi todo el código con la de crear. Una de las diferencias que tiene la vista de editar es que los campos que ya contienen información deben presentar esta misma al momento de editar la estructura.

Tabla 12, Semana 14

| Semana | Descripción | Logros |
|-----------|-----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Semana 14 | <ul style="list-style-type: none"> Se crearon el resto de las vistas para las estructuras. | <ul style="list-style-type: none"> Se comenzó el desarrollo de las vistas de crear y editar de las estructuras de datos. |

Fuente: Propia, 2020

Después se desarrolló la última vista de las estructuras que es la de mostrar. En esta vista se deben mostrar todas las llaves y valores de la estructura, las cuales deben venir de Redis. En el controlador se hace la conexión a Redis en la cual se piden todas las llaves pertinentes, y se transforman a un objeto paginable dentro de Laravel para facilitar la navegación a través de estas. Ilustración de la vista mostrando una estructura de ejemplo:



Ilustración 19, Vista de mostrar estructura

Fuente: Propia, 2020

Se crea un controlador de Redis con un punto final para manejar peticiones a llaves hash, uno para poder modificar llaves hash y uno para eliminar una llave en particular. Estos puntos finales se utilizaron en la vista de mostrar como opciones para cada llave como se puede ver en la ilustración previa. Toda esta lógica se desarrolló en Javascript para que se puedan llevar a cabo estas operaciones sin tener que salir de la página actual.

Por último en esta vista se agrega una barra de búsqueda para poder filtrar todas las llaves de acuerdo a una cadena de caracteres, la cual también es paginada apropiadamente.

Tabla 13, Semana 15

| Semana | Descripción | Logros |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| Semana 15 | <ul style="list-style-type: none"> Se comienza el desarrollo de la vista de mostrar para las estructuras de datos. | <ul style="list-style-type: none"> Se desarrollan las características principales de la vista |

| | | |
|--|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> Se crea un nuevo controlador para manejar ciertas llamadas a Redis. | <ul style="list-style-type: none"> de mostrar para las estructuras de datos. Se crearon puntos finales para poder hacer consultas a Redis. |
|--|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Fuente: Propia, 2020

Hasta este momento las vistas de las estructuras solo manejan las estructuras tipo hash, pero en la implementación de Redis se utilizan tres tipos de estructuras: las hash, los sorted sets, y una combinación de ambos que dentro de este proyecto se refieren como híbridos. En este momento se modificaron las vistas y el controlador para agregar soporte a los otros dos tipos de estructuras.

En el controlador de las estructuras se agregó la lógica que maneja los otros tipos de estructuras.

En las vistas de crear y modificar se agregó un campo llamado tipo de estructura el cual es una lista desplegable que contiene los tres tipos de estructuras disponibles. Dependiendo de la selección del usuario, el formulario de crear o editar cambia dinámicamente ya que el número de columnas de un sorted set y de un híbrido es fijo, dos y tres respectivamente, a diferencia del hash que puede llegar a contener más de tres columnas. Esto se puede visualizar en la siguiente captura que contiene los tres casos:

The image shows three screenshots of a web form for creating structures, each with a different type selected in the 'Type' dropdown:

- Hash:** The 'Type' dropdown is set to 'Hash'. Below it, there is a section titled 'Hash Columns' with four input fields, each labeled 'Enter column name'. The first field has a green 'Add' button, and the other three have red 'X' buttons.
- Sorted Set:** The 'Type' dropdown is set to 'Sorted Set'. Below it, there is a section titled 'Sorted Set Columns' with two input fields: 'numeric_value' and 'string_value'.
- Hybrid:** The 'Type' dropdown is set to 'Hybrid'. Below it, there is a section titled 'Hybrid Columns' with three input fields: 'lower_limit', 'upper_limit', and 'stored_value'.

Ilustración 20, Vista de creación de estructuras

Fuente: Propia, 2020

La vista de mostrar la estructura se difiere mucho entre los tres tipos de estructura diferente, por lo cual se creó una vista de mostrar para cada uno y dependiendo del tipo de estructura se muestra la vista respectiva.

En esta semana también se comenzó el desarrollo del manejo de usuarios. Se creó el controlador de usuarios así como la lógica de inicio de sesión y autenticación.

Tabla 14, Semana 16

| Semana | Descripción | Logros |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 16 | <ul style="list-style-type: none"> • Se agregó soporte para las estructuras Sorted Sets e Híbrido. • Se comenzó el desarrollo del manejo de usuario. | <ul style="list-style-type: none"> • Se modificaron el controlador y las vistas para agregar soporte a los otros tipos de estructura. • Se creó el controlador de usuarios y se agregó la lógica de manejo de sesiones. |

Fuente: Propia, 2020

Luego se continuó el desarrollo de los usuarios a los cuales se le agrega un campo de administrador en la tabla, el cual los administradores tendrán habilitado para tener acceso a la administración de estructuras y la administración de usuarios.

Se programaron los CRUD para los usuarios en el controlador. Y se desarrollaron las vistas que hacen uso de estas.

Se utiliza un middleware para agregar restricciones en las páginas que solo pueden ser vistas por un administrador y también se esconden los botones que llevan a estas si el usuario que está ingresado no es administrador. A continuación un diagrama que muestra la filtración de peticiones a través del middleware que revisa si el usuario es administrador:

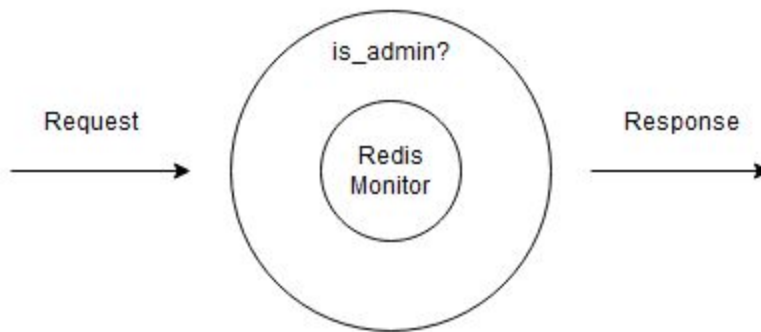


Ilustración 21, Diagrama de middleware

Fuente: Propia, 2020

También se agregó una restricción en todas las páginas, excepto la página de aterrizaje, para que solo puedan ser vistas por un usuario registrado.

Tabla 15, Semana 17

| Semana | Descripción | Logros |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 17 | <ul style="list-style-type: none"> • Se desarrollaron los CRUD para los usuarios. • Se desarrollaron las vistas de los usuarios. • Se agregó el usuario de administrador. | <ul style="list-style-type: none"> • Se terminó el desarrollo del controlador de usuarios. • Se desarrollaron las vistas de mostrar, crear y editar de los usuarios. • Se agregó la opción de administrador al momento de crear un usuario así como las restricciones para usuarios sin permiso en el resto de la aplicación. |

Fuente: Propia, 2020

Por último se agregaron dos páginas al controlador de páginas y sus enlaces en la barra de navegación para mostrar la salud del servicio de Redis. En una de estas se muestra información sobre el servicio de Redis, y en la otra se despliega la bitácora del servicio de Redis.

En la primera página se muestran estadísticas así como información sobre la instalación de Redis de forma ordenada. Esta información la provee Redis a través del comando info. Se agregó un navegador para acceder fácilmente a las diferentes categorías de la información.

En la segunda página se muestra la bitácora así como un campo de filtración por cadena de caracteres y un campo de límite de filas, que por defecto se encuentra en cien. Utilizando Javascript y CSS se destacan las ocurrencias de la consulta utilizando otro color.

También se agregó una revisión del estado del servicio de Redis cada vez que se carga una página dentro la aplicación. Si Redis no está disponible al cargar una página, la aplicación mostrará un error encima del contenido. Y por último se agregaron casos donde se maneja el fallo de las llamadas a Redis para que la aplicación no muestre errores graves si Redis no se encuentra disponible.

Tabla 16, Semana 18

| Semana | Descripción | Logros |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Semana 18 | <ul style="list-style-type: none"> ● Se agregó una página para mostrar información de Redis. ● Se agregó una página para mostrar la bitácora de Redis. ● Se consideró el caso de que Redis no esté disponible. | <ul style="list-style-type: none"> ● Se desarrolló una página para mostrar datos pertinentes al servicio de Redis. ● Se desarrolló una página para mostrar y navegar la bitácora de Redis. ● Se agregaron casos en los que Redis está caído y ahora se muestran errores relevantes. |

Fuente: Propia, 2020

VI. Conclusiones

Durante la práctica profesional se implementó Redis como base de datos clave-valor en memoria sobre el servicio principal de Movitext y la información previamente hardcoded se encuentra en Redis disponible a petición.

Se logró familiarizarse con las tecnologías utilizadas en ambos proyectos, entre estas PHP, Perl, Laravel y Redis.

Se diseñaron las estructuras de datos para almacenar la información en Redis.

Se desarrollaron nuevos plugins que consultan información en Redis.

Se desarrolló una aplicación web para administrar la información guardada en Redis utilizando el marco de trabajo Laravel.

VII. Recomendaciones

Existen diversas tecnologías y marcos de trabajo para el desarrollo de aplicaciones web y con cada una de estas una infinidad de conocimiento por adquirir. Al lidiar con tantas tecnologías a lo largo de mi carrera universitaria así como en el ambiente profesional, se recomienda a los estudiantes de la carrera indagar en la mayor cantidad de tecnologías y especializarse en la que encuentren más adecuada a sus habilidades.

A la empresa se le recomienda implementar algunas prácticas de marcos de trabajo ágiles, especialmente las reuniones diarias.

VIII. Bibliografía

- (2016). Introduction to EMG 7. Obtenido de EMG:
[https://doc.nordicmessaging.se/display/EMG/Introduction to EMG 7](https://doc.nordicmessaging.se/display/EMG/Introduction+to+EMG+7)
- (2020). Obtenido de Redis: <https://redis.io/topics/faq>
- (2020). Obtenido de Git: <https://git-scm.com/about/>
- (2020). Obtenido de Perl: <https://www.perl.org/about.html>
- (2020). Obtenido de Apache: https://httpd.apache.org/ABOUT_APACHE.html
- (2020). History of PHP and Related Projects. Obtenido de PHP: <https://www.php.net/history>
- (2020). Obtenido de Laravel: <https://laravel.com/docs/4.2/introduction>
- (2020). Where work happens. Obtenido de Slack: <https://slack.com/>
- (2020). History of MySQL. Obtenido de MySQL:
<https://dev.mysql.com/doc/refman/8.0/en/history.html>
- (31 de 01 de 2020). How To Install LAMP stack On CentOS 7 - Tutorials. Obtenido de Falconinternet:
<https://www.falconinternet.com/2020/01/31/install-mysql-apache-mysql-and-php/>
- (26 de 05 de 2020). Obtenido de ECMA-262: <https://tc39.es/ecma262/#sec-overview>
- Brown, P. J. (1985). *Writing interactive compilers and interpreters*. Wiley.
- Deka, G. C. (2017). NoSQL: Database for storage and retrieval of data in cloud. Obtenido de
<https://aws.amazon.com/nosql/key-value/>
- Freeman, A. (2013). The MVC Pattern. *Pro ASP.NET MVC 5*, 51-66.
doi:10.1007/978-1-4302-6530-6_3
- Montemayor Marin, M. C., Bijarro Hernandez, F., & Estrada Bellman, P. C. (2007). *POLÍTICAS Y GESTIÓN PÚBLICA PARA EL ESTUDIO MUNICIPAL: CONSULTORÍA ESPECIALIZADA*. Tamaulipas.

Monty. (27 de 01 de 2008). Sun buys MySQL AB. Obtenido de <http://monty-says.blogspot.com/2008/01/sun-buys-mysql-ab.html>

Naji, C. (21 de 09 de 2016). Obtenido de <https://www.justinmind.com/blog/4-prototyping-process-models-to-streamline-software-development/>

Oppel, A. J., Groff, J. R., & Weinberg, P. N. (2010). *SQL: The Complete Reference ; Third Edition*. McGraw-Hill.

Surguy, B. M. (24 de 01 de 2014). History of Laravel PHP framework, Eloquence emerging. Obtenido de <https://maxoffsky.com/code-blog/history-of-laravel-php-framework-eloquence-emerging/>

Ullman, J. D., & Widom, J. (1997). *A first course in database systems*. Prentice-Hall.

Wall, L., Orwant, J., & Christiansen, T. (2000). *Programming Perl, 3rd Edition*. O'Reilly Media.