



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

# **Creación de un CRM con el modelo SCRUM.**

**INFORME DE ACTIVIDADES PROFESIONALES**

Que para obtener el título de  
**Ingeniero en Computación**

**P R E S E N T A**

Edgar Tapia Ontiveros

**ASESOR DE INFORME**

Ing. Alberto Templos Carbajal



**Ciudad Universitaria, Cd. Mx., 2021**

# 1. Índice.

1. Índice .....	2
2. Nombre del proyecto.....	3
3. Objetivo .....	3
4. Marco teórico .....	3
5. Antecedentes del tema .....	4
6. Definición del problema o contexto de la participación profesional ...	4
7. Análisis y metodología empleada .....	6
8. Participación profesional.....	15
9. Resultados obtenidos .....	52
10. Mejoras al sistema a futuro .....	53
11. Conclusiones .....	54
12. Bibliografía.....	55

## **2. Nombre del proyecto.**

Creación de un CRM con el modelo SCRUM.

## **3. Objetivo.**

Crear un CRM funcional con el uso de diferentes herramientas de programación actuales y utilizando la metodología SCRUM.

## **4. Marco teórico.**

Hoy en día un Customer Relationship Management mejor conocido como CRM se usa en la mayoría de las empresas, su definición puede ser:

Un CRM se define como *“la integración de tecnologías y los procesos de negocios usados para satisfacer las necesidades de los clientes durante cualquier interacción con los mismos”* (Bose, 2002, p.89).

Es decir que el CRM se ocupa para interactuar con los clientes. Las aplicaciones que le podemos dar a un CRM son diversas y van a depender del sistema y sus limitantes que éste tenga, así como de las preguntas que nos hagamos y su posible solución que tratemos de darle con éste.

La creación del CRM presentado en este informe fue realizada mediante la metodología llamada SCRUM, esta metodología pertenece a las llamadas ágiles y es una de las más importantes y usadas hoy en día para la creación de proyectos.

Adicional a la metodología SCRUM, se realizó utilizando herramientas para el desarrollo de aplicaciones mediante componentes de interfaz de usuario, en especial los pertenecientes a DevExpress.

Y por último se aplicó la reutilización de software para reducir costos en general: *“La ventaja obvia de la reutilización de software es que los costos de desarrollo deberían reducirse.”* (Somerville, 2005, p. 381).

## **5. Antecedentes del tema.**

La empresa donde laboro compró aproximadamente en el año 2008 un CRM a una empresa con sede en Nueva York, Estados Unidos.

Con el paso de los años fue creciendo una gran inquietud de mejora de la herramienta, esto debido a que el uso de CRM tuvo un fuerte auge en las empresas por la necesidad de llevar un mejor control de sus clientes, el problema es que la herramienta con el paso del tiempo y con la llegada de más competencia se comenzó a hacer limitada en algunas de sus funcionalidades.

Estas limitantes y el peligro de pérdida de los clientes que se tenían llevó a cuestionarse el mejorar la herramienta creando un CRM completamente desde cero para solventar todas esas necesidades que se nos estaban presentando.

## **6. Definición del problema o contexto de la participación profesional.**

En el momento que entré a la empresa donde actualmente laboro, llegué como desarrollador de software y me fue encargado actualizar el CRM con las necesidades que se presentaban en ese momento.

Debo mencionar que esta responsabilidad que se me asignó fue complicada al principio, ya que el lugar donde actualmente laboro es mi primer y hasta la fecha único trabajo.

Se me dio un plazo de un par de meses para poder adaptarme a la herramienta y a las tecnologías en las que se trabajaban, afortunadamente tuve la ayuda de varios compañeros desarrolladores que trabajaban conmigo y que previamente tenían a cargo el CRM por lo cual el periodo de adaptación y el entendimiento de este fue más rápido de lo esperado. Otra cosa que ayudó bastante es que tenía conocimientos del lenguaje de programación utilizado el cual era C# y en general las herramientas de .NET.

Pasado este periodo de adaptación previamente mencionado, realicé todas esas actualizaciones solventando los problemas que se tenían en ese momento, sin embargo, con el paso del tiempo y con las necesidades creciendo más y más nos dimos cuenta de que esas modificaciones realizadas estaban haciendo que cada vez fuera más difícil de darle un mantenimiento adecuado a nuestro programa.

Poco tiempo después se logró determinar que la herramienta nos limitaba, en parte por su esquema multitenant debido a que toda la información de los clientes recaía en una misma

base de datos, por lo cual mantener el sitio y la base de datos costaba demasiado trabajo por la alta cantidad de registros por licencia de cada cliente.

Con esta problemática se decidió realizar el CRM desde cero aprovechando toda la experiencia que ya se tenía de la versión anterior del CRM.

Esta nueva versión del CRM me fue encargada junto con un equipo de trabajo debido a los buenos resultados que mostré previamente al actualizar la versión anterior y es el motivo de mi Informe Profesional.

Por motivos de confidencialidad con la empresa en la cual laboro, mostraré en el informe la información indispensable para que se pueda entender el trabajo sin aunar en detalles específicos del sistema que actualmente se vende en México y Estados Unidos.

## 7. Análisis y metodología empleada.

Para la realización de este programa nos fue asignado un tiempo de 6 a 10 meses aproximadamente, con la finalidad de realizar el CRM con la misma funcionalidad que tenía el anterior. Este periodo de tiempo se determinó metiendo una lista de tareas principales en el software JIRA y este a su vez nos brindó el estimado de tiempo para la realización del software.

Adicional a esto teníamos que tener en cuenta que debíamos dejar el CRM preparado para poder agregar algún módulo de manera rápida sin afectar la funcionalidad que ya estaba hecha y que los módulos del CRM no estuvieran amarrados entre sí, esto con el fin de que si había un bug o error afectara sólo el módulo y no el CRM por completo y nos fuera más fácil el darle mantenimiento.

Como inicio para realizar este proyecto decidimos emplear una gama de aplicaciones utilizadas por la empresa para la creación de los proyectos, esto principalmente porque se cuentan con las licencias para poder utilizar estos programas y contábamos ya con el conocimiento para hacer uso de ellas.

Como primera aplicación utilizamos una llamada JIRA la cual está desarrollada por la empresa ATLISSIAN y que según su página web:

*“Jira Software es una herramienta ágil de gestión de proyectos compatible con cualquier metodología ágil, ya sea scrum, Kanban o la tuya propia. Con funciones que van desde los tableros ágiles, los backlogs, las hojas de ruta y los informes hasta las integraciones y los complementos, puedes planificar, supervisar y gestionar todos los proyectos de desarrollo de software ágil con una sola herramienta. Elige una metodología para ver cómo Jira Software puede hacer que tu equipo publique software de mayor calidad con más rapidez.”* (JIRA, 2021).

Esta herramienta es una de las más famosas a nivel mundial debido a su fácil uso y como la descripción anterior lo dice prácticamente es compatible con cualquier metodología. Por lo tanto, en la empresa la ocupamos para mantener tanto el control de los proyectos, como el control de las personas y que se realicen de manera correcta y en el tiempo determinado las tareas asignadas a esta.

Otra de las herramientas que decidimos utilizar para realizar el proyecto fue otra aplicación también perteneciente a la empresa ATLISSIAN llamada Confluence la cual según su página web:

*“Podrás crear y organizar el trabajo en un solo lugar, así como colaborar en él, desde virtualmente cualquier sitio.”* (CONFLUENCE, 2021).

*“Confluence se ha diseñado específicamente para los equipos que necesitan una forma segura y fiable de colaborar en proyectos cruciales. Los sitios de Confluence están protegidos por controles de privacidad y cifrado de datos, y cumplen con los estándares de cumplimiento verificados por el sector.”* (CONFLUENCE, 2021).

Esta herramienta la usamos para crear un proyecto y gestionarlo, dentro de Confluence diseñamos el proyecto, damos de alta los objetivos, sus pros y contras de éste y los pasos a seguir que posteriormente daremos de alta en JIRA.

Otra de las herramientas utilizadas y que generalmente usamos en todos los proyectos que hacemos dentro de la empresa es DevExpress, la cual según su página web:

*“DevExpress es una herramienta que ofrece a los desarrolladores de aplicaciones una de las suites más completas de componentes de interfaz de usuario (UI) en todas las plataformas .NET tales como Windows Forms, MVC, ASP.NET, Silverlight y Windows XAML.*

*Incluye distintos componentes tales como tablas, calendarios, editor de HTML, Hojas de cálculo, editores de datos o gráficas.*

*Para poder utilizar DevExpress, es necesario tener una licencia, la cual, varía su costo en función del plan contratado. Cuando se adquiere, el desarrollador tiene acceso a un foro a través del cual podrá obtener soporte técnico, las librerías de cada uno de los componentes y el código fuente de los mismos, siendo posible llevar a cabo modificaciones de dicho código, en el caso de que el desarrollador lo necesitare.”* (DevExpress, 2021).

Esta herramienta se ocupa tal cual para programar, como la descripción lo dice cuenta con componentes que nosotros podemos usar en nuestros diferentes tipos de proyectos para hacer más fácil y rápido la creación de estos.

Una herramienta que también ocupamos y que forma parte de la familia de DevExpress, es la llamada DevExtreme la cual según su página web:

*“Desde Angular y React hasta Vue, DevExtreme incluye una colección completa de componentes de interfaz de usuario receptivos y de alto rendimiento para su uso en aplicaciones web tradicionales y aplicaciones móviles de próxima generación. La suite se envía con una cuadrícula de datos con todas las funciones, componentes de gráficos interactivos, editores de datos y mucho más. Con DevExtreme, brindará experiencias de usuario increíbles para los navegadores web modernos de hoy.”* (DevExtreme, 2021).

Esta herramienta la usamos para la compatibilidad de los navegadores que utilizamos hoy en día y para la adaptabilidad de las diferentes tecnologías que estos navegadores soportan.

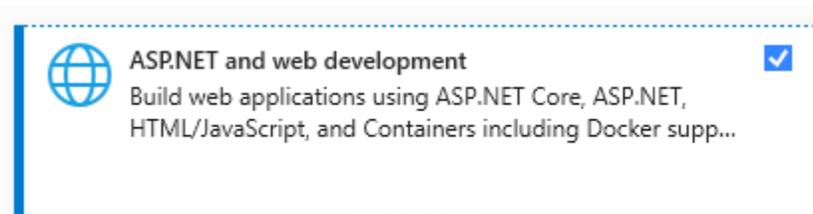
Otra herramienta que se utilizó para el envío de correos masiva es la llamada Mailchimp la cual según su página web:

*“Mailchimp ayuda a que las pequeñas empresas hagan grandes cosas, con las herramientas y pautas adecuadas en cada paso del camino.”*

*“Envía los mensajes correctos en los canales adecuados. Crea correos electrónicos, anuncios para redes sociales, páginas de destino, postales y mucho más, desde un solo lugar.”* (MailChimp, 2021).

En resumen, utilizamos esta herramienta para hacer el envío de campañas de correo a una cierta cantidad de contactos.

Se utilizó el programa Visual Studio 2019 como entorno de desarrollo para realizar el proyecto, con la característica principal de ASP.NET and web development.



**Figura 1. Característica de Visual Studio de ASP.NET.**

También ocupamos el programa SQL Server para poder realizar la base de datos a la cual iba acceder nuestro nuevo CRM. Se determinó utilizar SQL Server debido a la experiencia que teníamos con el programa y el CRM anterior, ya que la base de datos estaba realizada en SQL.

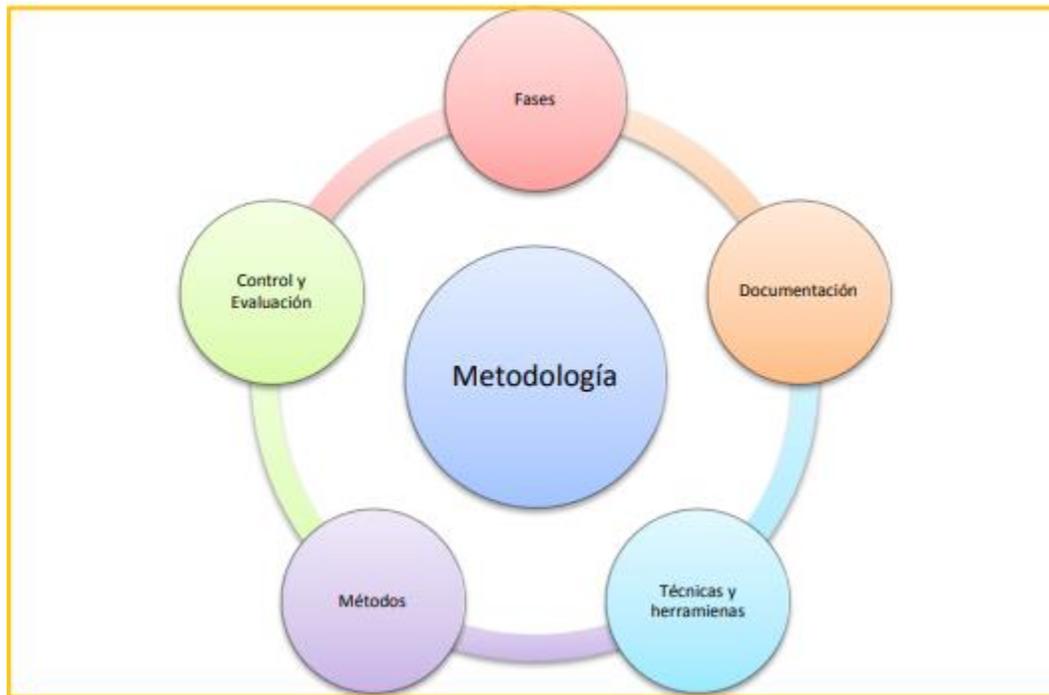
Como metodología usamos la llamada SCRUM. Primeramente, debemos decir que significa metodología y según varios autores es:

*“Definiremos la metodología como aquella disciplina que indicara que métodos y técnicas hay que usar en cada fase del ciclo de vida del desarrollo del proyecto.”* (UOC, 2021).

*“Metodología se refiere al estudio de los métodos como objeto de conocimiento. Es la teoría de los métodos que ordena las operaciones cognoscitivas y prácticas, en la acción racional profesional.”* (Lima, Boris, 1963).

*“Es el conjunto de operaciones o actividades que, dentro de un proceso preestablecido, se realizan de manera sistemática para conocer y actuar sobre la realidad social. Hace referencia a los supuestos epistemológicos.”* (Ander Egg, Ezequiel, 1982).

Los elementos que componen una metodología son:



**Figura 2. Elementos de la metodología.**

Ya entendiendo el concepto de metodología, podemos entonces adentrarnos a la utilizada la cual fue la SCRUM.

Existen diversas definiciones de esta metodología, pero una que a mi parecer engloba todo lo que representa es la dada por el sitio [proyectosagiles.org](http://proyectosagiles.org):

*“Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.*

*En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.*

*Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costos se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando*

*un proceso especializado en el desarrollo de producto.” (PROYECTOSAGILES.ORG, 2021)*

Adicional a la definición anterior debemos decir que SCRUM al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en SCRUM llevan el nombre de “Sprints”.

Para entender el ciclo de desarrollo de SCRUM es necesario conocer las 5 fases que definen el ciclo de desarrollo ágil.

1. Concepto: Se define de forma general las características del producto y se asigna el equipo que se encargará de su desarrollo.

2. Especulación: En esta fase se hacen disposiciones con la información obtenida y se establecen límites que marcarán el desarrollo del producto, tales como costos y agendas.

Se construirá el producto a partir de las ideas principales y se comprueban las partes realizadas y su impacto en el entorno.

Esta parte se repite en cada iteración/sprint y consisten en general en:

- Desarrollar y revisar los requisitos generales.
- Mantener la lista de las funcionalidades que se esperan.
- Plan de entrega. Se establecen las fechas de las versiones, hitos e iteraciones. Medirá el esfuerzo realizado en el proyecto.

3. Exploración: Se incrementa el producto en el que se añaden las funcionalidades de la fase de Especulación.

4. Revisión: El equipo revisa todo lo que se ha construido y se contrasta con el objetivo deseado.

5. Cierre: Se entregará en la fecha acordada una versión del producto deseado. Al tratarse de una versión, el cierre no indica que se ha finalizado el proyecto, sino que seguirá habiendo cambios denominados “mantenimiento”, que harán que nos acerquemos al producto deseado.



**Figura 3. Fases del Desarrollo Ágil.**

Ya conociendo el ciclo de Desarrollo Ágil podemos decir que SCRUM va a gestionar las iteraciones a través de reuniones de manera diaria, estas reuniones las podemos ver como:

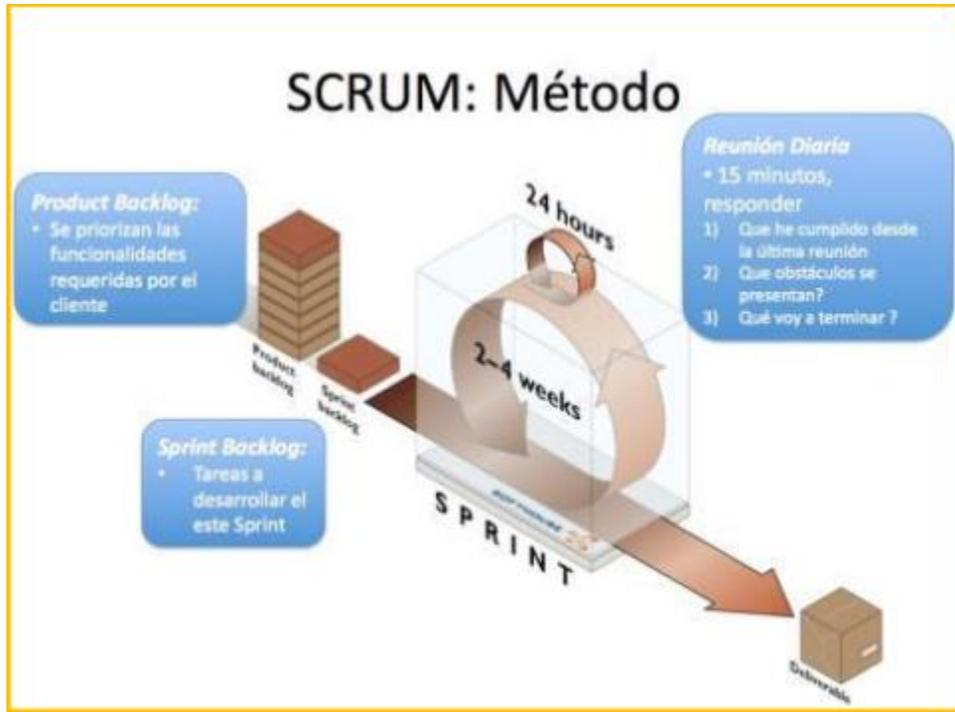
1. Planificación del Backlog: En estas reuniones se reflejan los requisitos del sistema por prioridades y son en donde se generan los Sprints, en el cual se deciden los objetivos y el trabajo para esa iteración. Además, en esta reunión obtenemos el Sprint Backlog, que es la lista de tareas a realizar y es el objetivo más importante del Sprint.

2. Seguimiento del Sprint: En esta fase se hacen preguntas diarias en las que se hacen 3 preguntas principales para evaluar el avance de las tareas, y son:

- ¿Qué trabajo se realizó desde la reunión anterior?
- ¿Qué trabajo se hará hasta una nueva reunión?
- Inconvenientes que han surgido y que hay que solucionar para continuar y terminar con las tareas.

3. Revisión del sprint: Cuando se finaliza el sprint se realizará una revisión del incremento que se ha generado.

A grandes rasgos podemos decir que la metodología Scrum funciona con los 3 puntos anteriores.



**Figura 4. Metodología SCRUM.**

Posterior al término del último Sprint y con los resultados obtenidos podemos decir que el producto está terminado.

Y para terminar de complementar el análisis de mi trabajo debo mencionar lo que son los requerimientos funcionales y no funcionales.

Requerimientos Funcionales. Son declaraciones de los servicios que proveerá el sistema.

Requerimientos No Funcionales. Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de este.

Listare una lista de mis requerimientos funcionales y no funcionales a tomar en cuenta en el desarrollo de mi sistema.

## Requerimientos Funcionales.

1. El sistema permitirá a los usuarios autorizados el dar de alta nuevas Empresas, Contactos, Oportunidades o Casos.
2. El sistema permitirá editar o borrar cualquier ítem perteneciente a los módulos principales.
3. A cada ítem creado dentro de cualquiera de los módulos se le dará un identificador único.
4. El sistema permitirá la creación de una actividad para registrar la interacción del usuario con el ítem con el que se esté trabajando.
5. El sistema debe tener la capacidad de realizar las búsquedas que el usuario necesite.
6. El sistema debe tener la capacidad de expirar la Licencia en la fecha que se le indique.
7. El sistema validara automáticamente el Usuario asociado a una Licencia en específico.
8. El sistema controlará el acceso y lo permitirá únicamente a usuarios autorizados. Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña.
9. El sistema podrá filtrar mediante los grid los diferentes tipos de datos, int, bool, char, etc.
10. El sistema podrá exportar la información de los grid si es necesario.
11. Cualquier intercambio de información vía internet se realizará por medio del protocolo encriptado https.

## Requerimientos No Funcionales.

1. Los permisos de acceso al sistema podrán ser únicamente cambiados por el administrador.
2. Las bases de datos se deben respaldar cada 24 horas. Los respaldos deben ser almacenados en una localidad segura ubicada en una ruta diferente a la que reside el sistema.

3. Si se identifican ataques de seguridad en el sistema, el mismo no continuara operando hasta ser desbloqueado por un administrador.
4. El sistema debe contar con manuales de usuarios estructurados correctamente.
5. El sistema debe presentar mensajes de error que sean informativos y orientados al usuario final.
6. El sistema debe poseer un diseño adaptativo al fin de garantizar la visualización en múltiples dispositivos.
7. El sistema debe poseer interfaces graficas bien formadas.
8. El sistema debe tener la capacidad para manejar diferentes lenguajes.
9. El sistema debe tener un plan de recuperación ante desastres.
10. La metodología de desarrollo del software será SCRUM.
11. El proceso de desarrollo se gestionará por medio de una herramienta web para gestionar el proceso de desarrollo de software.

## 8. Participación profesional.

Como bien lo mencioné primero utilizamos el programa Confluence. Debido a que era una de las personas que más conocían sobre el CRM me fue asignado y junto con otro par de personas dimos de alta todos los requerimientos dentro de Confluence, así como también anotamos las nuevas funcionalidades y los posibles pros o contras que se nos iban a presentar.

Como primer paso definimos el nombre de proyecto, para nuestro caso independientemente que fuera un programa hecho desde cero, nosotros lo manejamos como una nueva versión de nuestro CRM anterior por lo cual llevó el mismo nombre, pero con un número de versión diferente.

Posteriormente anotamos las personas involucradas en el proyecto, en este caso fuimos un grupo de 8 personas, 3 programadores, 1 diseñador, 1 persona involucrada directamente con los clientes para conocer que funcionalidades nuevas se requerían y por último 3 personas asignadas a llevar el seguimiento del proyecto. Estas últimas personas fueron nuestro CEO, nuestro director de CRM y nuestro director de Desarrollo los cuales junto conmigo estuvimos encargados de gestionar y hacer la tarea en algunos momentos como Project Manager.

Luego anotamos las tecnologías a utilizar las cuales mencioné en el Análisis y metodología empleada.

Ya definida la metodología y los programas que íbamos a utilizar, se anotó el tiempo de entrega el cual como comenté en el punto anterior fue de 6 a 10 meses.

Luego escribimos el objetivo el cual era realizar el CRM con mínimo la misma funcionalidad que tenía la versión anterior, más otros módulos que se requerían debido a las necesidades de nuestros clientes.

Con esto en mente anotamos la funcionalidad y la arquitectura que tenía el CRM anterior, la cual era:

1. Página web realizada en ASP.NET
2. Framework 3.5
3. Página con arquitectura multitenant
4. Módulo de manejo de Licencias.
5. Módulo de manejo de Usuarios.

6. Módulo de Empresas.
7. Módulo de Contactos.
8. Módulo de Oportunidades.
9. Módulo de Casos.
10. Módulo de Tareas.
11. Funcionalidad de Tareas en los módulos de Empresas, Contactos, Oportunidades y Casos.
12. Funcionalidad de Interacciones en los módulos de Empresas, Contactos, Oportunidades y Casos.
13. Envío de correos para notificar.
14. Importación de Empresas, Contactos, Oportunidades y Casos mediante un documento de tipo Excel.
15. Creación de Campos Custom en los módulos de Empresas, Contactos, Oportunidades y Casos.
16. Creación de Reportes para los módulos de Empresas, Contactos, Oportunidades y Casos.
17. Creación de catálogos para el uso de los módulos de Empresas, Contactos, Oportunidades, Casos y Tareas.

Luego se agregó la funcionalidad y la arquitectura que debía tener la nueva versión del CRM, la cual fue:

1. Página web realizada en MVC.
2. Utilizando los componentes de DevExpress y DevExtreme.
3. Utilizando Javascript.
4. Utilizar Bootstrap y hacer adaptable la página
5. Framework 4.5 ó superior.
6. Módulo de manejo de Licencias.

7. Módulo de manejo de Usuarios.
8. Módulo de Empresas.
9. Módulo de Contactos.
10. Módulo de Oportunidades.
11. Módulo de Casos.
12. Módulo de Tareas.
13. Módulo de Marketing.
14. Módulo de Documentos.
15. Módulo de Calendario.
16. Módulo de Reportes.
17. Funcionalidad de Tareas en los módulos de Empresas, Contactos, Oportunidades y Casos.
18. Funcionalidad de Comentarios en los módulos de Empresa, Contactos, Oportunidades y Casos.
19. Funcionalidad de Notas en los módulos de Empresas, Contactos, Oportunidades y Casos.
20. Funcionalidad de Actividad en los módulos de Empresas, Contactos, Oportunidades y Casos.
21. Funcionalidad de Documentos en los módulos de Empresas, Contactos, Oportunidades y Casos.
22. Envío de correos para notificar.
23. Apartado de notificaciones para observar cuando se asignaba una nueva tarea o se creaba una nueva entidad en cualquiera de nuestros módulos de Empresas, Contactos Oportunidades y Casos.
24. Importación de Empresas, Contactos, Oportunidades y Casos mediante un documento de tipo Excel.
25. Exportación de los datos pertenecientes a cualquier módulo.

26. Creación de Campos Custom en los módulos de Empresas, Contactos, Oportunidades y Casos.
27. Creación de Reportes para los módulos de Empresas, Contactos, Oportunidades y Casos.
28. Creación de catálogos para el uso de los módulos de Empresas, Contactos, Oportunidades, Casos y Tareas.
29. Creación de campañas mediante el módulo de Marketing utilizando la herramienta MailChimp.
30. Método de búsqueda más simple y moderno.
31. Borrado lógico de los registros de cualquier módulo.
32. Metodología Kanban integrada al módulo de Oportunidades.
33. Funcionalidad de Branch dentro del CRM
34. Creación de módulos externos de manera fácil.
35. Crear de manera rápida funcionalidad de Tareas en módulos externos.
36. Crear de manera rápida funcionalidad de Comentarios en módulos externos.
37. Crear de manera rápida funcionalidad de Notas en módulos externos.
38. Crear de manera rápida funcionalidad de Actividad en los módulos externos.
39. Utilizar un WebService de tipo REST para hacer toda la parte de consultas en la base de datos.

Después se agregó las características que se tenían en la base de datos con el CRM anterior, las cuales eran:

1. SQL Server 2012
2. Una única base de datos para todo el CRM con arquitectura multitenant

Las características que se determinaron usar para la base de datos con la nueva versión del CRM fueron:

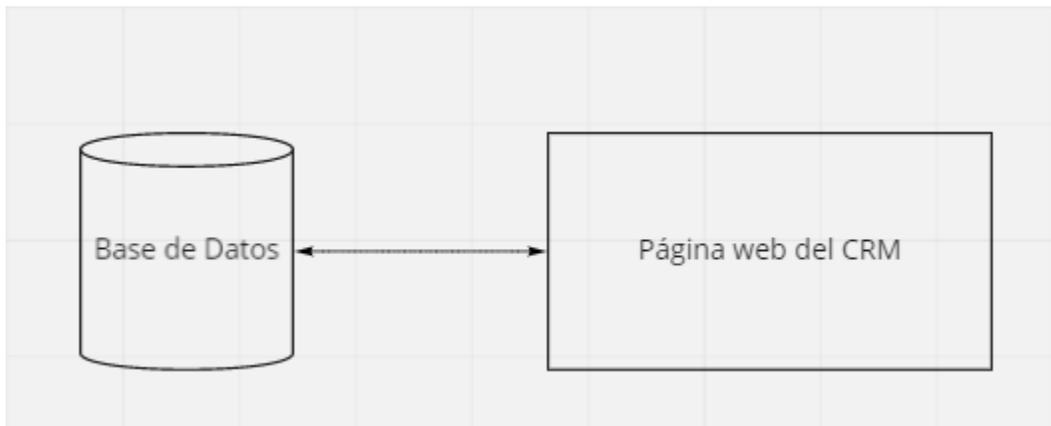
1. SQL Server 2016 ó superior

2. Base de datos para manejo de Licencias
3. Base de datos por Licencia
4. Quitar la arquitectura multitenant

Posterior a eso se agregaron diagramas, el primero es de cómo estaba la base de datos con el CRM anterior, así como su explicación.

Se utilizaba una base de datos multitenant conectada directamente al sitio web. Este tipo de arquitectura tiene como principal ventaja que es económica en desarrollo y mantenimiento, ya que al ser una única base de datos los costos se dividen entre los diferentes clientes.

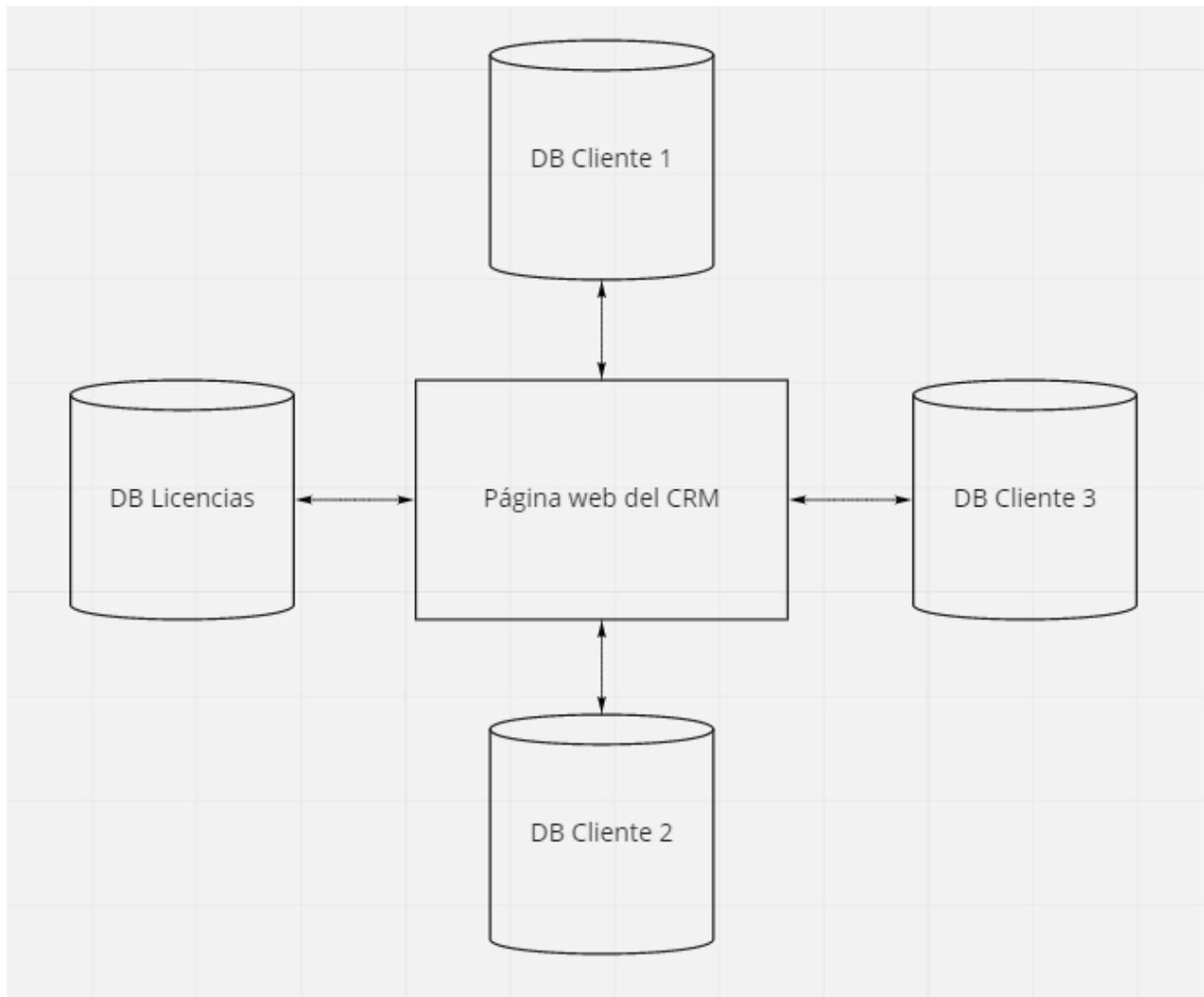
Sin embargo, debido al ser una única base de datos para todos nos generaba una gran desventaja al querer desarrollar características específicas para un único cliente, así como también si fallaba algo, fallaba para todos los clientes.



**Figura 5. Base de datos Multitenant conectada a un CRM. Esta base de datos tiene múltiples licencias con su data.**

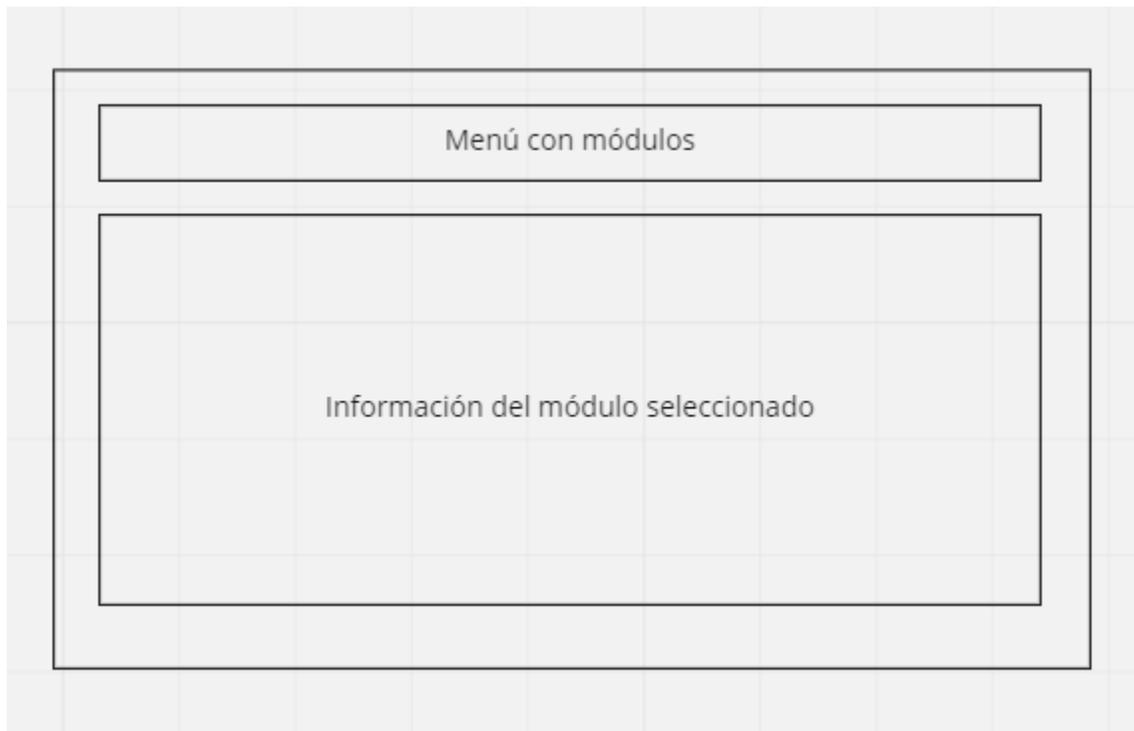
El segundo diagrama fue el del nuevo esquema de base de datos para la nueva versión del CRM.

Este nuevo esquema, tenía como principal diferencia que ya no era una arquitectura multitenant dentro de una única base de datos, contaba con una base de datos únicamente para el manejo de las Licencias y diferentes bases de datos por cliente con sus datos específicos, todos éstos a su vez conectados a un único sitio web.



**Figura 6. Base de datos de Licencias y bases de datos de Clientes conectadas a un CRM.**

Nuestro tercer diagrama era del diseño de la página web que manejaba el CRM anterior, en este caso era un menú superior con todos los módulos y debajo de él dependiendo lo que se eligiera cambiaba la información dentro de un contenedor en la pantalla, por lo cual esta página desperdiciaba demasiado espacio.



**Figura 7. Sitio web del CRM con espacio de la pantalla desperdiciado.**

En el cuarto diagrama creamos el nuevo diseño de la página web del CRM, este diseño tenía como fin ocupar todo el espacio de la página, así como hacerla adaptable a cualquier tamaño de resolución, incluso a un teléfono móvil. A su vez el menú superior iba contar con un menú de tipo hamburguesa, el cual se iba desplegar al momento de hacer clic.



**Figura 8. Sitio web utilizando el espacio completo de la pantalla. Menú de hamburguesa cerrado.**



**Figura 9. Sitio web utilizando el espacio completo de la pantalla. Menú hamburguesa abierto.**

Posterior a esto y teniendo toda la relación de cosas que íbamos hacer, revisamos las prioridades de cada punto, como mencioné anteriormente el objetivo principal era tener mínimo la misma funcionalidad que el CRM anterior, por lo cual partiendo de ello tomamos los puntos más importantes a realizar para lograr ese primer objetivo.

Por último, definimos lo que iba ser el esquema del Webservice tipo REST que íbamos a ocupar, decidimos que, aunque ocupáramos un modelo de tipo MVC, para nosotros iba ser más fácil crear un Servicio al cual hacerle peticiones OnDemand de la información, es decir que, si en un determinado momento quisiéramos saber la información de un módulo en específico podríamos ir a pedir la información al servicio y que éste a su vez la regresaría de manera rápida.

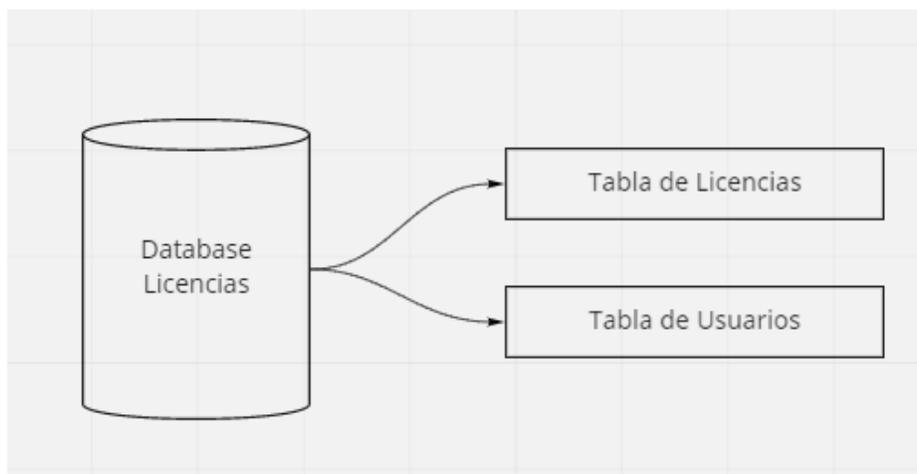
También se decidió de esta forma para tener dividida la parte de funciones y la parte web y que si fallaba una función no tener que modificar nuestro proyecto del CRM y sólo modificar el servicio.

Decidimos que el servicio iba contener los modelos para que no estuvieran dentro de nuestra aplicación del CRM e iba estar dividido en unas cuantas clases principales, las propias del servicio en donde iban estar las definiciones de las funciones que se iban a llamar, las cuales se decidió que podían ser de tipo de petición GET, POST, PUT y DELETE y otras clases en donde iban estar las funciones que iban a traer la información de la base de datos y que a su vez la iban a regresar de la forma en que nosotros queríamos.

Con esto en mente nos dispusimos a utilizar la herramienta JIRA para crear los Sprint y el Backlog dentro de éste.

La primera prioridad u objetivo para que todo pudiera funcionar era crear la base de datos con un diseño funcional que se acomodara a nuestras necesidades. Como primer responsable del Sprint #1 y sus tareas quedé yo y como lo mencioné antes lo primero fue el diseño y la creación de las bases de datos que íbamos ocupar para nuestro CRM.

Respecto al diseño de la base de datos, se creó una única base de datos para la manipulación de las licencias. Dentro de esta base creamos dos tablas, una con las licencias y otra con los usuarios asociados a la licencia. En la estructura de la tabla de Licencias entre uno de sus campos contenía si la licencia estaba activa o no, así como su tiempo de expiración, ya que el CRM se vende por Licencia y cantidad de Usuarios. En la estructura de la tabla de Usuarios se encontraba la LicencialD, con lo cual sabíamos a qué licencia pertenecía cada usuario y el correo que para nosotros es único, es decir que un correo no puede pertenecer a varias licencias. Se diseño de esta forma ya que en la base de datos multitenat llegamos a tener diferentes usuarios con el mismo correo.

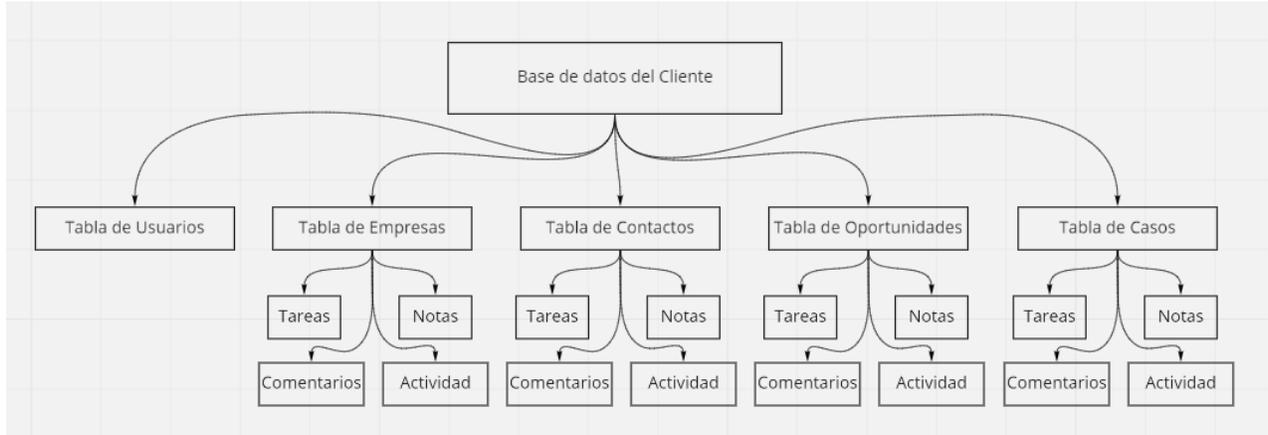


**Figura 10. Base de Datos de Licencias, cuenta con dos tablas, Licencias y Usuarios.**

Posterior a la creación de la base de datos de Licencias o Redirección como yo le llamo, me dispuse a crear la base de datos que íbamos a utilizar para el CRM, para esta base de datos ya teníamos la experiencia de la que usábamos para la versión anterior del CRM, sin embargo, como esa base de datos utilizaba una arquitectura de tipo multitenat no nos servía de mucho, por lo cual no pudimos hacer uso del mismo esquema de base de datos, pero si reutilizar muchas cosas de éste.

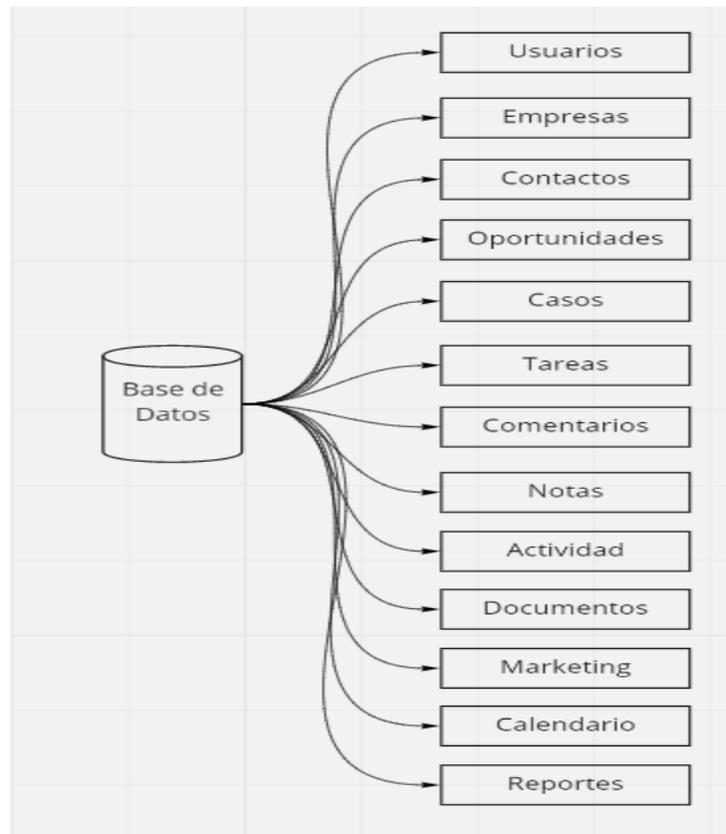
Como bien mencioné esta base de datos iba ser una por cliente, por lo cual nos bastaba con hacer una sola base de datos funcional y esa misma se iba replicar por cada cliente que tuviéramos.

Primero diseñé el esquema para ver como iba funcionar.



**Figura 11. Esquema de base de datos funcional.**

Con este esquema de uso, me dispuse a crear la base de datos con las tablas necesarias. Dentro de estas tablas debían estar las relaciones presentadas en el esquema anterior.



**Figura 12. Esquema de Base de Datos.**

Empezamos creando una base de datos de prueba y dentro de ésta para la funcionalidad básica metimos una tabla por módulo, así como diferentes tablas para Tareas, Comentarios, Notas y la Actividad, dentro de todas estas tablas metimos 2 campos principales los cuales fueron UserID y BranchID este último lo ocupamos para decir a que sucursal de la empresa pertenece, ya que en algún punto se nos presentó la necesidad dentro de la misma empresa como de algunos de los clientes que tenían varias sucursales en diferentes estados o países indicar a que sucursal se le iba crear esa nueva oportunidad o contacto.

Debo comentar que la tabla de Usuarios entre todos los campos debía contar con el correo como usuario, este correo debía ser el mismo que el que estaba en la base de datos de Redirección.

Las tablas de Tareas, Comentarios, Notas y Actividad contaban con los Id correspondientes de cada módulo para poder tener su relación obligada con éstos.

Después de diseñar la base de datos, crearla y terminar todas las tareas de ese Sprint #1 que aproximadamente duro 1 semana, volvimos a reunirnos para crear el Sprint #2.

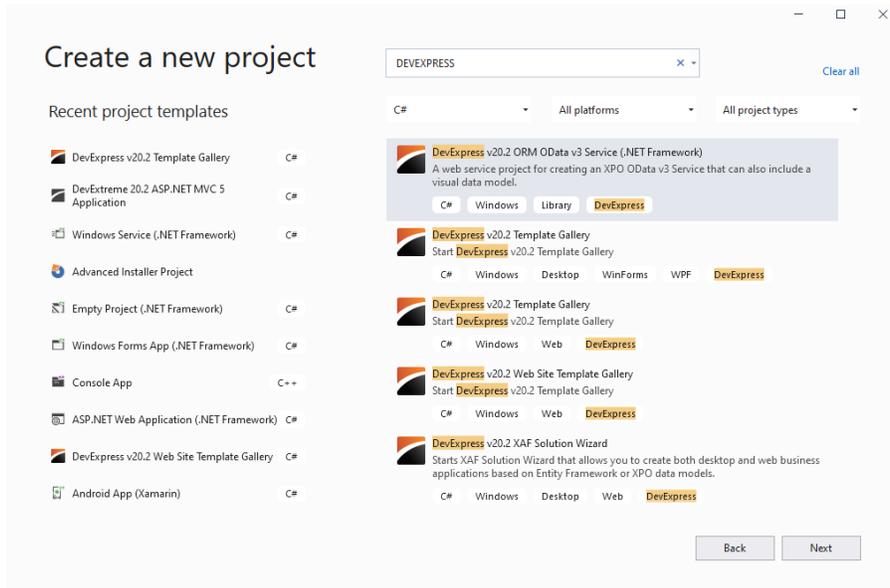
Tengo que destacar que como bien mencioné al explicar la metodología SCRUM, durante todos los Sprint todos los días nos reuníamos para ver si íbamos cumpliendo con los objetivos del Backlog para el Sprint correspondiente en cada semana. Durante los primeros 2 ó 3 Sprints no hubo mucho problema porque el que tenía que entregar las cosas era yo y de cierto modo no hubo tanta programación en ello, sin embargo, para los Sprints siguientes cuando se integraron los demás programadores y personas involucradas en el proyecto hubo algunas veces en que esos Sprints se atrasaron para completarse y con este atraso obviamente los tiempos de entrega subieron un poco.

Para el Sprint #2 decidimos que empezaríamos creando la página web y el servicio que íbamos ocupar. Iniciaríamos haciendo la búsqueda de información de cómo crear el servicio y la página web mediante las tecnologías elegidas para que se tuviera el diseño presentado en las figuras 8 y 9 que previamente ya habíamos planeado al diseñar el proyecto y darlo de alta en Confluence.

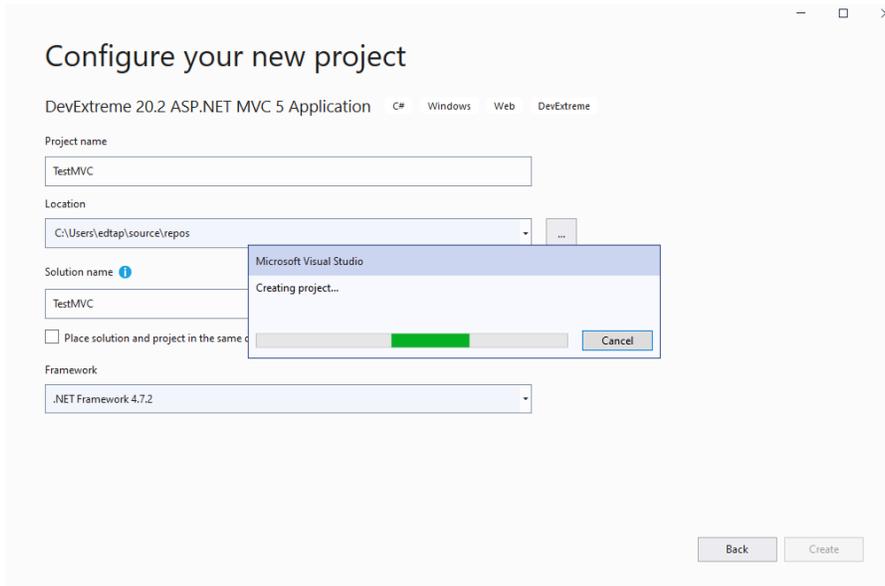
Para hacer la página web decidimos utilizar el Modelo Vista Controlador (MVC) utilizando Visual Studio, Javascript y los componentes de DevExpress y DevExtreme.

Este Sprint #2 también me fue asignado a mí por lo cual comencé creando el proyecto de la página web en Visual Studio, afortunadamente como DevExpress es un framework bastante amigable, pude crear un proyecto de tipo MVC utilizando un template de DevExpress, este template lo que hace de manera automática es crearte la estructura del proyecto que vas a usar y que en este caso fue de tipo MVC.

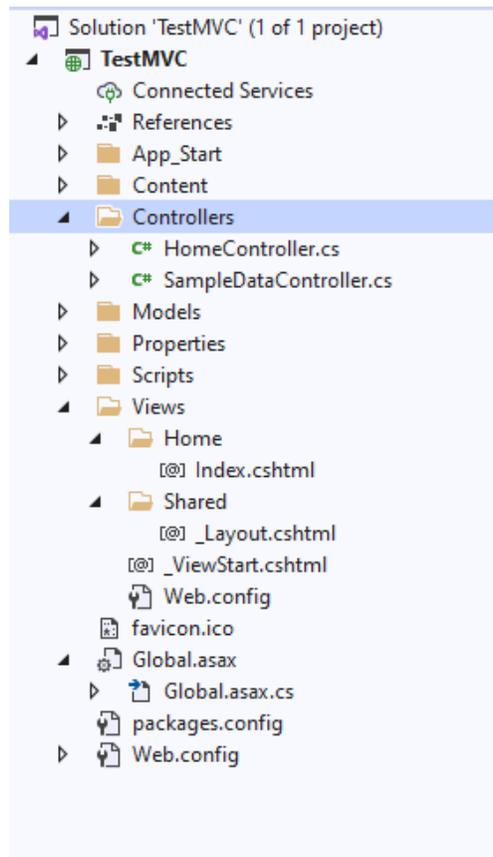
La estructura de este tipo de proyectos MVC cuenta principalmente con un Controlador, en el cual van a ir definidas las páginas que se van abrir, cada definición tiene su propia vista.



**Figura 13. Opciones para crear un proyecto desde Visual Studio con DevExpress.**



**Figura 14. Configuración y creación de un proyecto de tipo MVC de DevExpress desde Visual Studio.**



**Figura 15. Estructura de un proyecto creado desde el template de DevExpress.**

```
public class HomeController : Controller {  
    public ActionResult Index() {  
        return View();  
    }  
}
```

**Figura 16. Controlador con definición de vista llamada Index.**

Respecto al servicio también utilicé Visual Studio para crearlo y ahí no fue mucho problema ya que para este Sprint sólo lo creé y busqué información de como implementar las funciones en éste.

Con todo esto realizado se concluyó sin problemas la segunda semana y mi Sprint #2.

Algo que no mencioné antes es que cada viernes las personas involucradas de darle seguimiento al proyecto nos reuníamos para ver los avances durante la semana, darle fin al Sprint de esa semana si es que las tareas dentro del Backlog estaban terminadas y si era el caso que no estaban terminadas, terminar el Sprint de la semana y poner esas tareas pendientes para el Sprint de la siguiente semana.

Para el Sprint #3 asignamos un tiempo de 4 a 5 semanas aproximadamente y debido a que ya teníamos el esquema de la base de datos creado, estuvimos en condiciones para que pudieran empezar a intervenir los demás programadores, así como también nuestro diseñador, con ello en mente pudimos crear una lista de tareas más grande. A uno de ellos se le asignó la parte de Kanban para el módulo de Oportunidades. Su trabajo consistía en investigar el Modelo Kanban y ver cómo y con qué herramientas podíamos integrarlo. Al otro programador se le asignó empezar a ver la parte de importación de los registros en los módulos de Empresas, Contactos, Oportunidades y Casos.

En cuanto a mí, las tareas que se me asignaron fueron crear la parte del login, así como comenzar a crear las páginas para cada uno de los módulos. Para estas tareas trabajé mucho a la par con nuestro diseñador.

Primero comencé haciendo la página de Login, inicié con la parte visual creando primero los controles y acomodándolos como lo habíamos planeado anteriormente, tengo que comentar que en general para realizar todas las páginas utilicé los controles de DevExtreme los cuales trabajan con las tecnologías más nuevas de hoy en día y que son Javascript, React, Angular, Vue, etc. Esta página de Login carece de la funcionalidad de registrarse, esto debido a que la licencia se vende con los usuarios previamente ya creados, por lo cual los elementos con los que tenía que contar el login únicamente eran:

- Usuario
- Contraseña
- Botón de Login
- Botón de Recuperar Contraseña
- Logo de la Empresa
- Links de Política de Privacidad y Términos de Servicio

Con la página creada y los elementos definidos lo que hice fue pasársela al diseñador para que le diera la forma necesaria y poder dejar una buena presentación.

Después del front-end de la página de Login, seguí con la página para la Recuperación de la contraseña del usuario o como bien la llame Password Recovery, con esta página la funcionalidad que necesitábamos era que el cliente pusiera su usuario y presionara un botón para recuperar su contraseña.

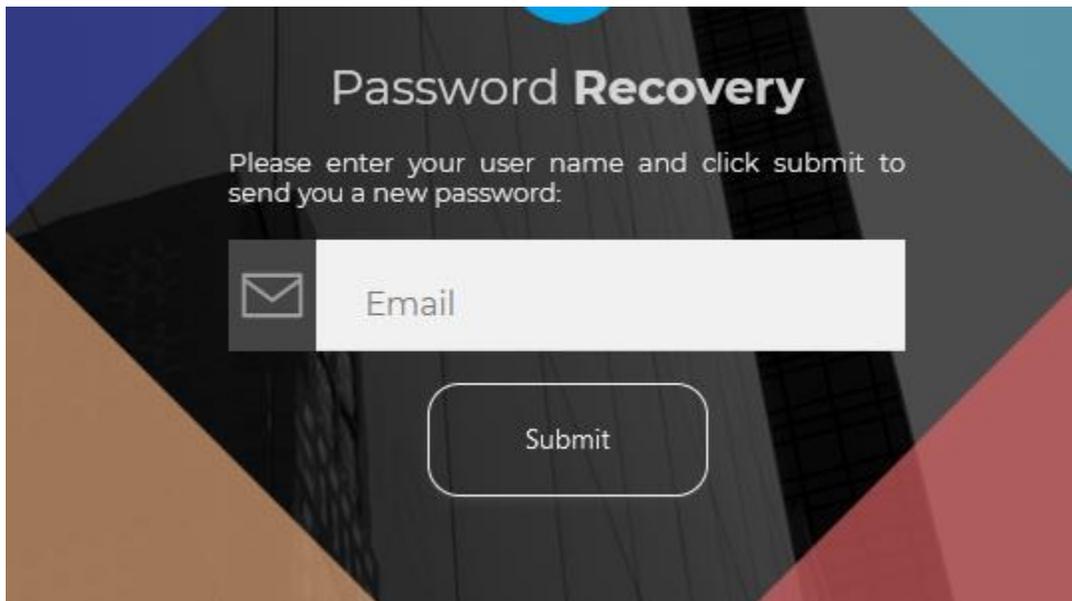
Lo que hice primero fue crear la página y usé controles de textbox y button pertenecientes a DevExtreme para la recuperación del usuario. Después lo que hice y que generalmente se hizo en todo el proyecto fue crear un javascript por página para poder mandar funciones desde ahí, generalmente por convención decidimos que el javascript se llame como el nombre de la página o Vista, en este caso el javascript se llamó Recovery.js.

Antes de usar mi archivo Recovery.js creé en mi servicio un modelo Recovery el cual constaba únicamente del correo y una función de tipo PUT la cual lo que hacía era primero recibir el correo a recuperar, buscaba en la base de datos de Redirección dentro de la tabla de Usuarios ese correo y con ello obteníamos a que licencia pertenecía ese correo. Posterior a ello se recuperaba todas las credenciales necesarias para acceder a mi Base de Datos a la que pertenecía esa licencia y ya en la tabla de usuario del cliente sacábamos la contraseña.

Ya con la contraseña recuperada lo que seguía era mandar un correo al cliente, indicándole su contraseña. Para el envío de correo se realizó desde una dirección de correo empresarial y se utilizó una clase realizada en C# creada por nosotros la cual hacía el envío de los correos.

Después de crear la función de Recovery dentro del servicio ahora sí tenía que pasar a mi Recovery.js, lo que hice fue crear una función que se ejecutaba al presionar el botón de recuperar contraseña, en esta función primero recuperábamos el valor del textbox, que en este caso fue el correo y con ello mandábamos a llamar la función PUT que se encontraba en mi servicio, esa función devolvía un true y un mensaje indicando que se había recuperado la contraseña o un false y el mensaje por el cual la contraseña no pudo recuperarse.

Si la contraseña se pudo recuperar, el textbox de la página Recovery se quitaba y se sustituía por un texto donde pusimos un mensaje indicando que la contraseña se pudo recuperar y fue enviada al correo del cliente, si no el mensaje debía decir que no se pudo recuperar la contraseña y que se comunicara con un administrador.



**Figura 17. Página de recuperación de contraseña.**

Terminada la página del Recovery se la pasé al diseñador para que le diera el diseño correspondiente, después de esto seguí con las páginas de errores, las clásicas que dan en una página web, Error 404, Error 500, Error 503, etc.

Para estas páginas de errores la verdad es que fue sencillo, hice la modificación a mi proyecto para que si en algún punto daba un error de este tipo me redirigiera a la página correspondiente y con esto poder saber qué tipo de error nos dio, en dónde y así poder realizar de manera rápida las correcciones pertinentes.

Después de hacer las páginas de errores se las pase al diseñador para que les diera su diseño correspondiente.

Ya que tenía la página de Recovery y todas las de Error, volví a la página de Login pero esta vez al back-end. Antes de comenzar mi back-end lo primero que hice fue crear un usuario en mi base de datos con todas sus características necesarias para hacerlo de tipo Administrador y poderlo usar para crear nuevos usuarios.

Ya teniendo mi registro dado de alta en la base de datos me puse a hacer mi back-end, primero utilicé mi botón de Login y le hice una función en mi Controlador de Login, en esta función lo que se hace es recuperar el usuario y la contraseña, luego el siguiente paso es ir a la base de datos de Redirección a validar si el usuario existe y si la licencia esta activa, si no está activa la licencia o el usuario no existe avisar y pedirle al usuario que se contacte

al administrador, pero si lo están lo que sigue es recuperar las credenciales necesarias para acceder a la base de datos del cliente, dentro de esta base de datos lo primero es validar que el usuario este activo, si no lo está se debe avisar al cliente que el usuario no está activo y que contacte a su administrador, pero si está activo lo que sigue es recuperar todos los datos del usuario necesarios para poder acceder y navegar en la aplicación, después de todas estas validaciones y que todas sean de forma correcta lo que resta es hacer la redirección a la página de Home.

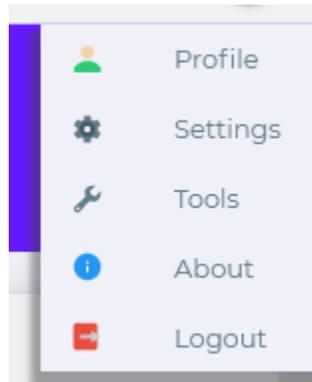
Entrando a la página de Home, en ese momento la única cosa que hice fue crearla en blanco y no hacerle ninguna modificación.

Después de que el Login quedó completo y me redirigía a mi página de Home, lo siguiente fue modificar el Layout de mi aplicación, en MVC podemos decirle a las páginas que carguen con un Layout en específico, este Layout funciona como una MasterPage por lo cual mi Layout lo que hacía es poder utilizarlo en todas mis páginas.

Dentro del Layout creé primero la barra superior, dentro de ésta como ya se había decidido se iba poner el branch al que pertenecía ese usuario, la imagen del usuario y el menú de hamburguesa. Para el menú hamburguesa se iban a desplegar toda la lista de menús a la que podíamos acceder según ciertos permisos que definiéramos en el módulo de Usuarios, para el branch al que pertenecía el usuario se iba agregar un botón que al presionarlo me iba dar la lista de branches a los que podía acceder y al seleccionar uno de estos elementos de la lista nos iba cambiar de branch junto con toda su información, para la imagen de usuario también se agregó un botón el cual al presionarlo nos iba desplegar una lista, en donde iban estar botones de:

1. Perfil de Usuario
2. Configuración
3. About Us
4. Logout

Todo el espacio restante lo deje para el contenido donde las demás páginas se iban a mostrar, esto como lo definido en las figuras 8 y 9.



**Figura 18. Opciones de Usuario.**

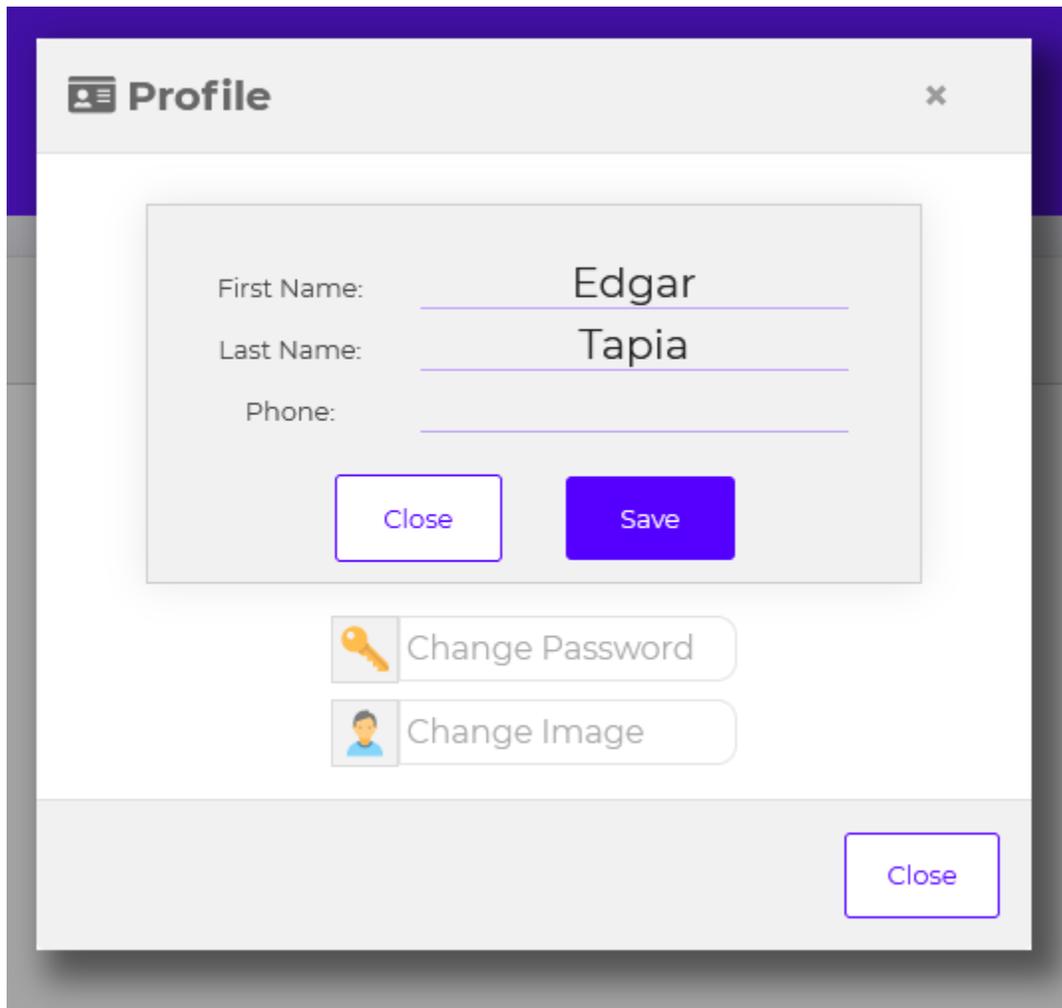
Primero se crearon dos pop-up, el primero fue para el About Us donde sólo puse el logo del CRM y la versión en la que nos encontrábamos, en el logo al presionarlo nos dirigía al sitio web del CRM en donde podíamos ver más a detalle la información sobre éste.

Mi segundo pop-up fue para el Perfil del Usuario, la funcionalidad de este pop-up constaba de 3 partes. La primera tenía los datos principales del usuario logueado los cuales son nombre, apellido y teléfono. La segunda parte era para cambiar la contraseña. Y por último la tercera parte era para cambiar la imagen del usuario del logueado.

Para el punto 4 que fue el de logout lo que se hizo es dentro del controlador de Login agregar una función para que hiciera el logout, esta función lo que hacía era cerrar todas las sesiones de usuario y cerrar la sesión del sitio web, esto con el fin para que ninguna sesión se mantuviera abierta.

Para el punto 3 que fue la configuración lo que hice fue crear una vista desde mi controlador, al crear una vista desde el controlador lo que se hace es que se crea la vista dentro de la carpeta vistas y se agrega la función en el propio controlador. Únicamente creé la página para posteriormente poder usarla.

Después de haber quedado los dos pop-up lo que hice fue pasárselos al diseñador para que él los adaptara y les pusiera el mismo diseño con el que contaba toda la página.



**Figura 19. Pop-up de perfil de Usuario.**

Luego me dispuse a hacer la parte del menú, lo primero que hice fue crear un menú de hamburguesa, primero le agregué los botones dentro del Layout, hasta este momento no le había agregado nada de funcionalidad, ésta la iba agregar dependiendo de los perfiles del usuario logueado, únicamente puse todos los botones con todos los modelos para que pudiera ver los módulos.

Posteriormente y a cada botón le definí que iba a pertenecer a un controlador, por ejemplo, para el módulo de Empresas lo que hice fue crear un controlador llamado Empresas dónde quedara exclusivamente la funcionalidad para este módulo, lo mismo pasó para el módulo Contactos y así sucesivamente para los demás módulos.

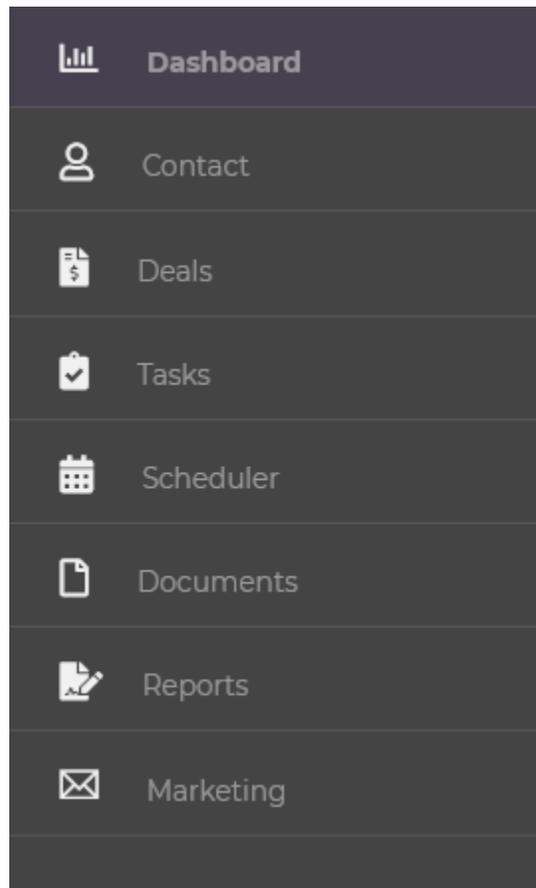
Por defecto al momento de crear cada controlador de manera automática se crea la función Index de cada uno de éstos, para esta función a su vez se le crea una vista principal asociada también llamada Index dentro de una carpeta llamada como el nombre del controlador.

Es decir, cada controlador tiene en la carpeta de Views una llamada como el propio controlador y dentro de ésta, una vista llamada Index. Por ejemplo, el controlador de Empresas tiene su propia Vista, es decir Views/Empresas/Index.

Ya creados mis controladores y vistas, hice un javascript por cada Controlador creado dentro de la carpeta de Scripts, esto para poder identificar de manera fácil y rápida que archivo modificar si en algún punto sucedía un error en algún módulo en específico. El script por controlador quedo así Scripts/NombreControlador/NombreControlador.js

Teniendo todo el esqueleto de mi programa con una estructura igual para todos los módulos con el fin de hacer más sencillo el manejo y creación de las cosas en cada uno de éstos, pasé a la parte del Layout nuevamente, en específico del javascript para el Layout, dentro del cual primero hice las funciones para manipular los elementos de la barra superior antes mencionados.

Para estos elementos primero se hizo la creación de las funciones dentro del javascript para abrir los pop-up al presionar los botones respectivos, luego se hizo la funcionalidad para cada uno de los módulos dentro del menú de hamburguesa. Como ya teníamos definidos las vistas y sus controladores, realizamos la funcionalidad para que al presionar el botón del módulo se redirigiera a la vista de éste, siempre conservando el mismo Layout.



**Figura 20. Menú de hamburguesa desplegado.**

Con todo esto y después de hacer las revisiones necesarias semana por semana terminé de forma correcta mi Sprint #3.

Posterior al término del Sprint #3 llego mi Sprint #4 en el cual íbamos a empezar a darle ya una funcionalidad completa al usuario, haciendo que los módulos se abrieran dependiendo de los permisos de usuario y comenzando la funcionalidad de algunos de los módulos.

Primero empecé con la página de Configuración, anteriormente esta página ya la había hecho al crear el Layout, sin embargo, no le había metido funcionalidad alguna, esta página fue creada dentro de un controlador de Administración y también tenía su javascript de Administración, este controlador iba llevar toda la funcionalidad de Administración dependiendo del usuario.

Empecé creando una especie de tarjeta en donde puse un botón de Configuración de Usuarios que me dirigían a una página de Usuarios creada, dentro de esta página hice toda la Administración de Usuarios en la cual podíamos decidir si un usuario iba pertenecer a uno o más branches y dentro de ésta misma creamos los pop-ups necesarios para poder agregar los datos de un nuevo usuario, o hacer la edición de este.



**Figura 21. Página de Configuración.**

Entre estos datos de usuario definimos los módulos, por lo cual al dar de alta el nuevo usuario decidíamos a que módulos se podían acceder. Cabe mencionar que estos datos de un nuevo usuario o edición de uno existente se iban a guardar a una base de datos por lo cual, a la vez de hacer estos cambios dentro de la página de Administración de Usuarios, fui creando las funciones HTTP en mi servicio que iban a ser llamadas mediante una función de tipo AJAX como anteriormente lo mencioné.

A la vez que iba creando esta administración uno de los otros programadores creo un árbol por jerarquía de usuarios en javascript dentro del cual con solo arrastrar al nodo de otro usuario podíamos decir que el jefe de ese usuario era el nodo seleccionado.

Al terminada esta parte del árbol jerárquico junto con el programador hicimos esa integración en mi proyecto para tener todo dentro de un mismo proyecto y no tener problemas después para juntar las cosas, con esto creamos un acceso directo desde mi Administración de Usuarios a este árbol y añadimos las funciones correspondientes al mismo servicio.

Después de tener el módulo de Usuarios me dispuse a volver al Layout, dentro de este tenía ya mi botón con el Branch seleccionado, por lo cual dentro de mi javascript del Layout hice la funcionalidad necesaria para hacer el cambio de Branch, al hacer el cambio de Branch se recarga toda la información asociada a ese Branch.

Luego de modificar el botón de Branch modifiqué mi menú de hamburguesa, en el cual por botón puse una validación que consistía en validar si mi usuario tenía el permiso para mostrar el módulo, si lo tenía el botón se mostraba de lo contrario permanecía oculto y sólo podía aparecer si se le daba acceso desde el módulo de Usuarios.

Debo mencionar también que a la hora de estar haciendo estas modificaciones otro programador estaba realizando una parte de diccionario, el cual por default el idioma iba ser inglés. Lo estábamos programando poniendo todas las etiquetas en inglés, sin embargo, con este diccionario podíamos cambiar el idioma de estas etiquetas al que nosotros decidiéramos.

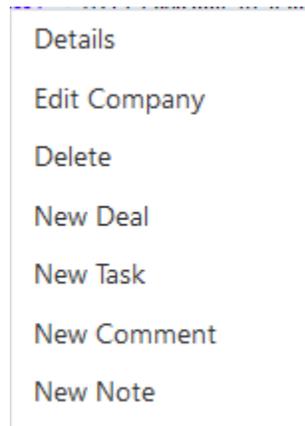
Dentro de este Sprint también realizamos esa integración del diccionario para cambiar las etiquetas que se fueran creando conforme íbamos programando, obviamente en cuanto fuera creciendo el programa nosotros íbamos ir agregando más etiquetas al diccionario.

Después de esta integración empecé la funcionalidad del módulo de Empresas, este módulo en todos los CRM es uno de los más importantes ya que trae la información de la Empresa a la que pertenece un Contacto, dentro de la información podemos destacar el nombre de la empresa, su teléfono, su correo, su información fiscal, su información de redes sociales, etc.

Antes de hacer la funcionalidad del módulo de Empresas, definí un esquema específico que iba utilizar para cada módulo, el cual iba ser un grid principal donde se listarán todos los registros asociados a este módulo, todos estos registros iban a depender del Branch al que pertenecían y del Usuario que los había creado, estos registros se iban a mostrar dependiendo la jerarquía de usuarios, es decir que si un empleado dio de alta ciertos registros, su rama hacia arriba es decir su jefe inmediato y las personas que estuvieran arriba de este también podrían ver ese registro que se dio de alta.

Para este grid utilice DevExtreme el cual nos ayuda bastante para la búsqueda de los usuarios dentro de este mismo grid, a su vez también ya trae incluida la funcionalidad de paginado y de la cantidad de registros para mostrar por página, también cuenta con la funcionalidad para ocultar ciertos campos y para seleccionar ciertos registros, o todos o algunos de ellos.

Dentro del grid pusimos un context menú también perteneciente a DevExtreme, en el cual se definieron varios botones, los cuales por módulo iban a tener más o menos las mismas opciones, pero todos iban a tener la opción de crear una nueva tarea, un nuevo comentario, una nueva nota o borrar el registro seleccionado.



**Figura 22. Context menú de un registro del módulo de Empresas.**

Ya definido este esquema básico por módulo comencé a crear el módulo de Empresas, primero hice la función dentro de mi servicio para traer las Empresas y ponerlo dentro de mi grid, como lo mencioné anteriormente estos registros se iban a traer por Branch y Usuario.

Después hice el botón de New en el cual íbamos a crear un nuevo registro de Empresas, se realizó el pop-up con su funcionalidad necesaria y se creó también la función para salvar esa nueva empresa en la base de datos.

Luego de crear la funcionalidad para una nueva empresa, hice la parte de edición de la empresa en la cual realicé también una función en mi servicio para traer los datos de un registro en específico, estos datos se cargaban y llenaban mi pop-up de edición que tenía las mismas características que el de nueva empresa, hay que comentar que también tuve que hacer la función dentro del servicio para hacerle update al registro que se estaba editando.

Companies

### New Company ✕

Name \*

Description \*

More

Tax ID	<input type="text" value="Enter TaxID"/>	Tax Name	<input type="text" value="Enter Tax Name"/>
Tax Address	<input type="text" value="Enter Tax Address"/>	Phone	<input type="text" value="🇺🇸 ▼"/>
Skype	<input type="text" value="Enter Skype"/>	Website	<input type="text" value="Enter Website"/>
Twitter	<input type="text" value="Enter Twitter"/>	Facebook	<input type="text" value="Enter Facebook"/>
LinkedIn	<input type="text" value="Enter LinkedIn"/>	Heat	<input type="text" value="Hot ▼"/>
Source +	<input type="text" value="Select... ▼"/>		

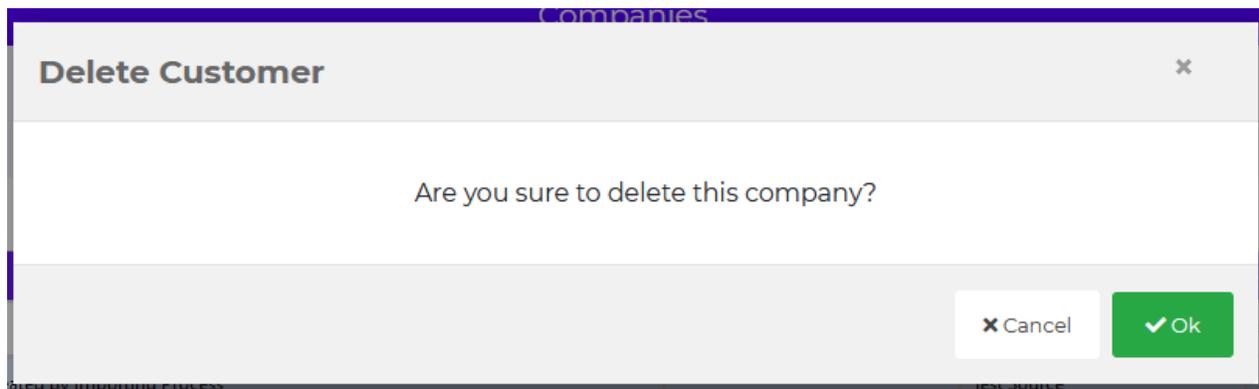
---

Informacion Cuenta

**Figura 23. Pop-up de Nueva Empresa.**

Ya teniendo la funcionalidad para nueva empresa y edición de una, me dispuse a crear la funcionalidad para el botón de borrar registro, como mencioné antes la función de borrado iba ser un borrado lógico por el cual se creó un pop-up que preguntaba si estaba seguro de borrar el registro, si lo estaba se activaba la funcionalidad para el borrado lógico en la base de datos, de lo contrario no se hacía nada.

Debo mencionar que el borrado lógico consiste en borrar el registro o dejar de mostrarlo dentro de mi grid pero en la base de datos el registro se queda junto con toda su información asociada a éste pero sin mostrarse, esto con el fin de que no se pierda por ningún motivo información alguna y que si es necesario se pueda recuperar de manera rápida cambiando únicamente el valor de borrado dentro de la base de datos.



**Figura 24. Pop-up de borrado de una Empresa.**

Luego de mi borrado lo que siguió fue crear mi pop-up para crear una nueva tarea en el cual la tarea va asociada a un usuario dentro de mi CRM, con cierta categoría y que debe ser efectuada en una fecha en específico. Para las tareas pusimos una fecha poder decidir la fecha de inicio, pero no la de fin, sin embargo, dentro de la base de datos si guardábamos la fecha en la que alguien completaba la tarea para tener el registro de cuánto tiempo tardo en completarse esa tarea y si era necesario poder utilizarla para algún reporte o gráfica. Generalmente en los CRM se define el tiempo en que sea va a hacer una tarea, sin embargo, lo que sucede casi siempre es que las tareas no se terminan en el tiempo específico que se determinó sino mucho tiempo después, por lo cual nosotros decidimos sólo dejar el día y la hora en que debe iniciarse y sólo las gráficas de la duración de las tareas se las íbamos a dar a un administrador del personal para ver si eran productivos o no.

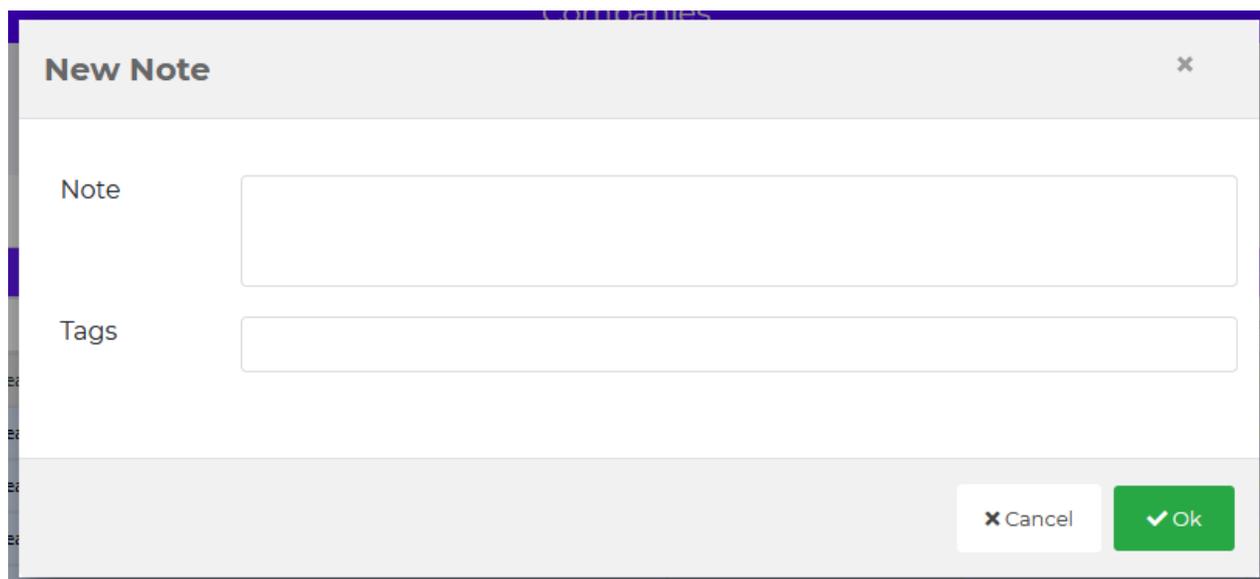
Para la categoría de las tareas uno de los programadores me ayudó haciendo la funcionalidad para poder agregar una categoría de tarea, ahí las categorías dependen de la propia organización, el administrador debe dar de alta que categorías van a tener, pero por default pusimos una de tipo regular.

Con esta administración de categorías creada lo que restaba era integrarla a mi módulo de Administración creando una tarjeta para tareas, la cual también se iba mostrar si el usuario tenía o no permisos para ver tareas.

Después de crear la tarea pase a crear el pop-up de Comentario, el cual consistía en agregar un comentario asociado a mi ítem seleccionado, este comentario únicamente era de tipo texto y al igual que todo iba guardarse por Usuario y Branch y en este caso también por Empresa. Para esto también se creó una función en el servicio para hacer el guardado.

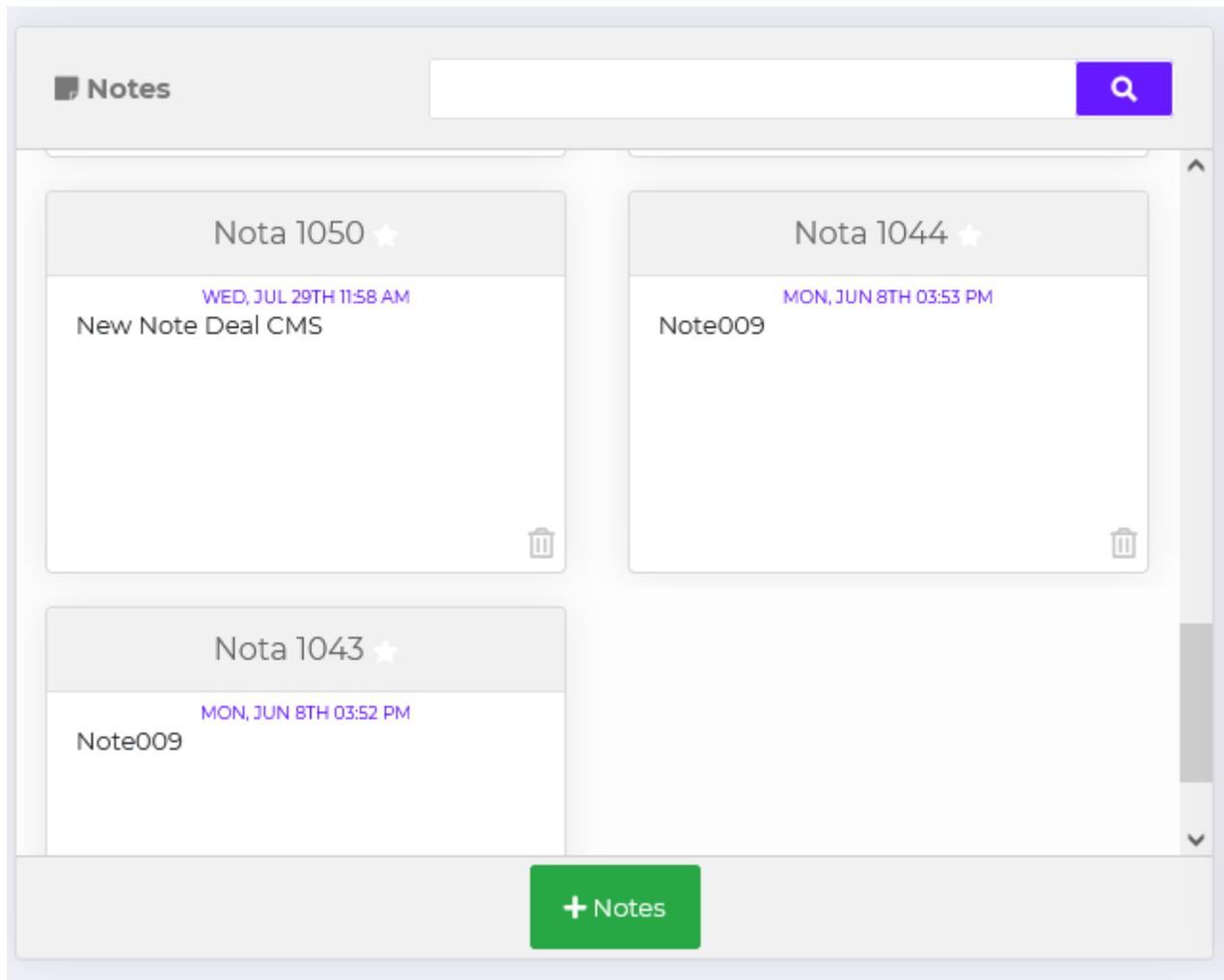
Terminando el pop-up de Comentarios solo nos faltaba crear el pop-up para las Notas, para estas uno de los otros programadores también me ayudó a crearlo. A las notas les quisimos dar una visualización tipo post it, que definimos como algo rápido que quieres que no se te olvide, pero en nuestro caso asociado al ítem seleccionado.

Se agregó la parte de las notas, sin embargo, si tuve que hacer algunos ajustes para acomodar a como queríamos las notas y dejarlas al igual que los demás elementos fáciles de usar y replicar en todo lo demás.



The image shows a screenshot of a web application's 'New Note' pop-up window. The window has a light gray header with the title 'New Note' and a close button 'x' in the top right corner. Below the header, there are two input fields: 'Note' and 'Tags'. The 'Note' field is a large, empty text area. The 'Tags' field is a smaller, empty text area. At the bottom right of the window, there are two buttons: a white button with a black 'x' icon and the text 'Cancel', and a green button with a white checkmark icon and the text 'Ok'.

**Figura 25. Pop-up para agregar una nota.**



**Figura 27. Vista de las notas creadas.**

Debo mencionar que cada que se agrega alguno de este tipo de ítems en el sistema nosotros también creamos de manera automática un registro en la base de datos, específicamente en la tabla de Actividad, esto para tener todo el histórico de lo que se hizo con ese ítem, desde quién lo creo, hasta quién agrego una tarea o quién borro el propio ítem. Se decidió hacerlo de esta forma porque en el CRM anterior llegamos a tener problemas de borrado o actualización de ítems y todos estos registros no quedaban guardados, entonces no sabíamos quién los habían hecho y en las diferentes organizaciones llegamos a tener casos que las personas se echaban la culpa entre ellos antes de asumir la responsabilidad por miedo a un regaño.

Por último, agregué una página de detalles en donde se mostraban la información básica de mi Empresa y mediante un control de tabs puse grids con todos los registros asociados a mi Empresa, en este caso se puso una tab para Tareas, otra para Comentarios, otra para Notas, otra para la Actividad realizada y posteriormente para todos mis módulos se agregó una tab de Documentos.



**Figura 28. Vista de detalles de una empresa.**

Al terminar de agregar toda esta funcionalidad y de cierto modo tener un módulo completo terminé el Sprint #4, creo que fue uno de los Sprint si no es que el más pesado ya que en el creé toda la funcionalidad que posteriormente iba replicar en los demás módulos.

Luego de concluido mi Sprint #4 pase a realizar el Sprint #5 el cual iba constar de los módulos de Contactos, Oportunidades y Casos.

Primero empecé creando la funcionalidad del módulo de Contactos que en un CRM es el más importante de todos los módulos, debido a que toda la interacción recae en un Contacto, esto independiente si pertenece o no a una empresa, que generalmente pertenece.

Para mi módulo empecé con el grid y su función en el servicio para traer la información de los contactos, luego hice los pop-ups de nuevo y edición, dentro de éstos entre su información básica están el nombre del contacto, apellido, teléfono, correo, redes sociales, etc. Y adicional a éstos se puso el nombre de la Empresa mediante un combobox, dentro de este combo nosotros podíamos o no seleccionar a que empresa iba pertenecer mi contacto. Luego, al igual que en el módulo anterior se pasó a la parte del Borrado y a la creación de los pop-ups de Tareas, Comentarios, Notas y por último a toda la parte del detalle.

Hay que mencionar que para la parte de creación de estos módulos se reutilizó mucho del código anterior, además algo que olvide comentar fue que dentro de la página del módulo en específico solo definíamos en el HTML los divs en donde iban estar asociados los controles de DevExtreme y en el javascript de cada módulo íbamos a tener la definición

de cada uno de estos controles que se iban a pintar en el HTML al momento de ser llamados, con esto también lo que hicimos fue que no todos los controles se cargaran al abrir la página, sino que se cargaran en el momento en que se pedían para que nuestra página fuera más rápida. Adicional a estos javascripts propios por módulo hice también un javascript global, en el cual nosotros poníamos la definición general de los controles que se iban a llenar con lo definido en los javascript propios de cada módulo y éste iba dar la orden de poner los controles en los módulos.

Después de terminado mi módulo de Contactos, le agregue el pop-up al módulo de Empresas, esto con la intención de poder crear un contacto asociado a una empresa desde mi módulo de Empresa.

Como mencioné anteriormente uno de los programadores estaba trabajando la parte de importación de Empresas, Contactos, Oportunidades y Casos, para este punto ya tenía el importador para estos dos módulos, ese importador partía de tener un documento de tipo Excel en donde estuvieran los registros a importar y lo que se hacía era hacer match con unos campos que ya estaban definidos, estos campos definidos eran los campos que tenía la forma de Nuevo de mis respectivos módulos, por lo cual sólo se hacía la relación con los campos que se fueran o no a ocupar. Con todo esto el importador creaba los registros en la base de datos y nosotros mediante todo lo que ya había realizado podíamos visualizarlos.

Para esta parte del importador únicamente lo que hice fue agregar un botón dentro del módulo de Empresas y otro en el de Contactos, con el cual al picarle nos abría en una nueva ventana ese programa que realizó ese programador.

Después de tener la importación de Empresas y Contactos me dispuse a crear mi módulo de Oportunidades, dentro de este módulo hice lo mismo que los dos anteriores, reutilicé la estructura que previamente había ocupado para toda la parte del grid y los módulos de Tareas, Comentarios y Notas, así como también la parte de detalles.

Para los pop-up de nuevo y edición en especial para esta parte de Oportunidades le pedí a uno de los programadores me ayudará a hacer unos catálogos los cuales se agregaron en la parte de Administración mediante una tarjeta, estos catálogos fueron para los ciclos de ventas y los pasos de estos ciclos de ventas, así como también los estatus para estas Oportunidades. Esto se hizo basándonos en la experiencia previa del CRM anterior y en la experiencia que se tenía en ese momento, con esto podemos decir que una oportunidad de ventas puede tener varios pasos para poder completarla, por ejemplo, los pasos para una venta sencilla pueden ser:

1. Contacto inicial

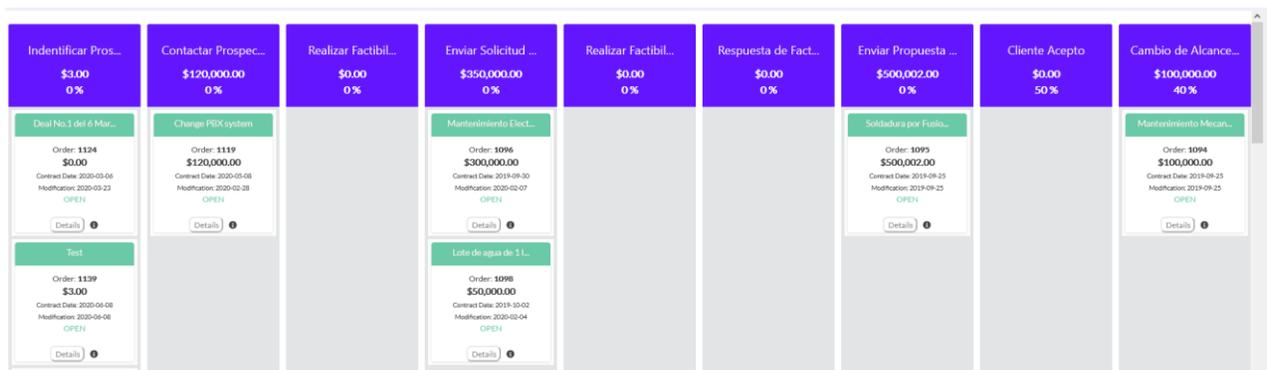
2. Análisis de requerimientos
3. Elaboración de propuesta
4. Presentación de propuesta
5. Negociación
6. Contratación
7. Instalación
8. Cobro
9. Contrato de mantenimiento

Cada uno de estos pasos dentro del ciclo de ventas lleva un porcentaje de éxito que va del 0 al 100 y tiene asociado un status que entre los más comunes puede ser:

1. Abierto
2. Cerrado ganado
3. Cerrado perdido
4. Rechazado
5. Stand By

Posteriormente y volviendo a mi pop-up de Nuevo, agregué un grid dentro del cual listaba los contactos que podían estar solos o asociados a una empresa, así como también la información básica que es el nombre de la oportunidad, su descripción, la elección de su ciclo de ventas, la moneda en la que se va a hacer y los ingresos estimados con esta oportunidad.

Luego de terminar de hacer todo esto, uno de los programadores me paso la parte del Kanban que mencioné anteriormente, la funcionalidad de este Kanban iba de la mano de mi ciclo de ventas, por lo cual se me mostraba mis ciclos de ventas y podía cambiar de paso solo con arrastrar mi tarjeta de Oportunidad a otro de los pasos. Integramos sin problema esta funcionalidad, poniendo un botón que me cambiaba a la vista de Kanban y a esta vista de Kanban puse otro botón que me cambiaba a la vista de grid.



**Figura 29. Vista de Kanban.**

Para finalizar mi módulo de Oportunidades pase la funcionalidad de crear una nueva Oportunidad a mis módulos de Contactos y Empresa, así como le agregue al detalle el grid con las Oportunidades asociadas respectivamente.

Posterior a ello me dispuse a crear mi módulo de Casos, el módulo de casos al igual que el de Oportunidades lleva un ciclo de casos por lo cual uno de los programadores me ayudo realizando este ciclo de casos, este al igual que el de ventas contaba con pasos con un porcentaje de ganancia y un status dependiendo el paso en el que nos encontrásemos.

Al igual que el módulo anterior se dio de alta el ciclo de casos en una tarjeta dentro de mi módulo de Administración que sólo se podía visualizar siempre y cuando mi usuario tuviera permisos para ver los Casos.

Después de creado utilicé como en los módulos anteriores la misma estructura, debo mencionar que al definir esta estructura básica para todos los módulos ya era muy sencillo crear un módulo nuevo, por lo que este último módulo no me llevo tanto tiempo como los primeros. Primero se creó la parte del grid y luego los pop-ups correspondientes a Tareas Comentarios y Notas, mi página de detalle y por último hice la parte de los pop-up de nuevo y edición los cuales al igual que en el módulo de Oportunidades tenían un grid en donde listaba los contactos y estos contactos se mostraban si estaban o no asociados a una empresa.

Hay que mencionar que entre todo esto se fueron creando las funciones dentro de mi servicio el cual ya tenían una estructura definida para poder agregar o editar cada tipo de cosa, por ejemplo, para las tareas, a la hora de mandar a llamar la función se definió pasar el módulo al que pertenecía y el ítem al que se quería agregar la tarea. Esto mismo lo hice con cada una de las funciones tratando de utilizar una misma estructura para poder reutilizar la mayoría de ellas y no hacer una por módulo.

Ya agregado lo necesario para que mi módulo de Casos funcionara de manera correcta, uno de los programadores me paso la parte de importación de Casos y Oportunidades y se integró. Con esta última integración ya podíamos importar registros a cualquiera de mis 4 módulos principales.

Con esto di por finalizado mi Sprint #5 y se planeó el Sprint #6 el cual iba contener la parte del módulo de Tareas, el módulo de Marketing y mi funcionalidad de Custom Fields. Los Custom Fields son campos que no están predefinidos en el sistema pero que el cliente necesita para el buen manejo de sus procesos.

Comencé el Sprint #6 mientras otros de los programadores empezaron haciendo el módulo de Documentos, la integración con MailChimp para mi módulo de Marketing y la creación de Reportes.

Primero hice el módulo de Tareas, en el cual como los módulos anteriores contaba con un grid principal donde listaba las tareas asociadas a este usuario, luego al pop-up de nueva tarea y de edición de tarea, que eran similar a los que estaban en los otros módulos les agregamos una opción más para tareas dirigidas a un grupo.

Hay que mencionar que nosotros definimos que las tareas van dirigidas de manera uno-uno o uno-muchos, es decir de uno a otro usuario o bien de un usuario a un grupo.

Para las tareas de usuario a grupo hice un catálogo en el módulo de Administración para dar de alta grupos, dentro de estos grupos definimos que usuarios iban a pertenecer a cada uno.

Volviendo a mi módulo de Tareas, en específico al pop-up ya teníamos la opción de dar de alta una tarea asignada a otro usuario, pero asignada a un grupo no estaba por lo cual hice la opción para crearla dirigida a un grupo, dentro de la opción para las tareas dirigidas a un grupo puse dos opciones:

La primera opción fue una tarea asignada a un grupo por el método de Round Robin el cual consiste en asignar la tarea a un usuario y seguir asignando dependiendo la pila de usuarios, hasta volver nuevamente al usuario en la posición 1 de mi pila.

La segunda opción consistía en crear una tarea dirigida al grupo, pero sin que tuviera un dueño, por lo cual la tarea les iba aparecer a toda la lista de usuarios de mi grupo y uno de esos usuarios tenía que tomarla y trabajarla.

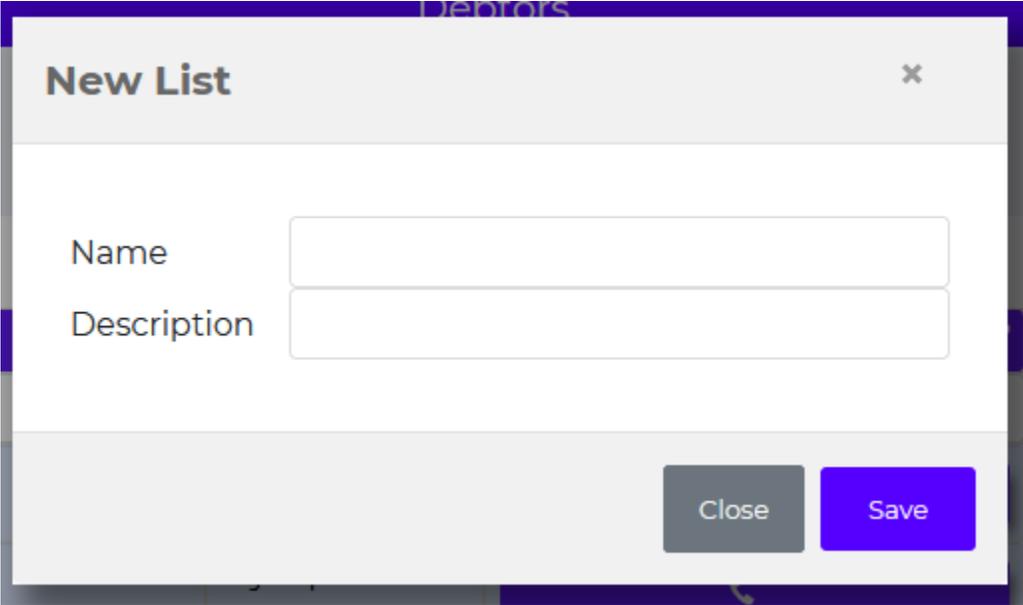
Después de generar esta opción en mis pop-up de nuevo y edición, replique esta funcionalidad en mis demás módulos que contaban con la opción de agregar una nueva tarea.

Luego de haber replicado la funcionalidad volví al módulo de Tareas al cual le agregué un botón que servía para abrir un pop-up que tenía un grid dentro donde se mostraban las tareas grupales que no habían sido tomadas y le metí la funcionalidad para poder tomarla y que ese usuario se hiciera su propietario y atendiera este tipo de tareas. Esta funcionalidad de grupo la tomamos basándonos en la experiencia con un prospecto que se tuvo, entre varias cosas que hacia este prospecto en su proceso de trabajo era dar soporte a sus clientes, por lo cual si surgía algún inconveniente con algún cliente daban de alta tareas en un repositorio general para que uno de los agentes la tomara, le diera seguimiento y la completará en un determinado tiempo.

Con esto completé mi módulo de Tareas y comencé a trabajar en el módulo de Marketing, para este módulo lo primero que hicimos fue crear 3 vistas principales, una para las Campañas, otra para Listas y una más para un diseñador HTML.

Comencé con la vista de Campañas, la cual al igual que todos los demás módulos contaba con un grid principal, en el cual se iban a mostrar las campañas creadas en MailChimp. Antes de continuar hay que mencionar que para crear una campaña en MailChimp se necesita de una lista y un HTML que es el que se va a mandar en la campaña de distribución masiva.

Luego hice la vista de Listas en donde se iban a mostrar también en un grid principal todas las listas creadas. Una lista sólo se podía llenar con Contactos por lo cual fui a mi módulo de Contactos y agregue un botón para crear una lista, después le agregue la funcionalidad que consistía en abrir un pop-up en el que podía crear una lista a partir de ciertos contactos seleccionados en mi grid o bien agregar contactos seleccionados a mi lista.



The image shows a modal window titled "New List" with a close button (X) in the top right corner. The form contains two input fields: "Name" and "Description". At the bottom right, there are two buttons: "Close" and "Save".

**Figura 30. Pop-up para la creación de una nueva lista.**

Por último, hice mi módulo de diseñador de HTML el cual consistía en un grid con la lista de los HTML creados, a su vez hice la funcionalidad para crear o editar un HTML, para este diseñador utilice un control de DevExtreme llamado HTML Editor, en el cual por defecto podemos crear un HTML y adicional a esto se le agrego la funcionalidad de guardado en la base de datos.

Con mis tres vistas creadas volví a la de Campañas en donde agregue el pop-up para crear una campaña, para realizar éste pop-up previamente uno de los programadores había ya investigado como integrar el MailChimp por lo cual se tuvo que crear una tabla en la base de datos para que ahí se dieran de alta los usuarios de MailChimp, con esta tabla hecha lo que restaba era crear algunas funciones en mi servicio las cuales utilizaban la API de MailChimp para crear, enviar y obtener la data de las campañas enviadas. Con esto creado complementé mi pop-up de una nueva campaña y quedó listo mi módulo de Marketing.

Para finalizar mi Sprint me enfoqué en la parte de los Custom Fields, como mencioné anteriormente son campos que no vienen incorporados en el sistema en ninguno de mis módulos principales, pero que sin embargo son necesarios para que el cliente pueda tener un mejor manejo en sus procesos, debo mencionar que esta definición de Custom Fields la creamos desde la versión anterior del CRM por la necesidad de brindarle al cliente mayor flexibilidad a la hora del manejo de sus procesos y no limitarlo únicamente a los campos que nosotros definimos.

Partiendo de esto, hice un catálogo de administración de mis Custom Fields, éstos los creaba dependiendo mis módulos principales los cuales son: Empresas, Contactos, Oportunidades y Casos y caían debajo de una Custom Tab, prácticamente la funcionalidad de la Custom Tab era agrupar ciertos Custom Fields en un espacio. Al momento de darlos de alta junto con la ayuda de otro de los programadores agregamos la funcionalidad en los pop-up de nuevo y edición de cada módulo para que les apareciera su apartado de Custom Tab con los Custom Fields dentro y pudieran dar de alta los campos correspondientes.

Para el momento en que terminé esto, uno de los programadores terminó la parte de Documentos, todo esto lo hizo en un proyecto y una base de datos aparte de la mía, viendo los pros y contras que esto nos llevaba y para tener sólo una única base de datos de administración de usuarios decidimos que las tablas de la base de datos debían ir creadas en mi base de datos que estaba ocupando para el CRM, por lo cual tuvimos que agregar esas tablas y funciones para poder tener la funcionalidad de documentos. El proyecto tal cual de documentos quedó aparte y para facilidad nuestra agregamos un módulo de Documentos en el cual embebimos mediante un iframe el proyecto previamente creado, pero que apuntaba directamente a nuestra base de datos.

A su vez también creamos una tab de Documentos para cada uno de mis módulos principales, esto con la finalidad de tener la facilidad de agregar documentos en cada ítem creado. Hay que comentar que esto se decidió para que se pudieran agregar cualquier tipo de documentos a un ítem, sobre todo a las Oportunidades y este tipo de documentos podían ir desde una cotización, una orden de compra, una requisición, una factura, etc.

Con esto finalicé mi Sprint #6 y estaba preparado para mi Sprint #7 en el cual hice mi Calendario y se agregó la funcionalidad de Reportes.

Para este Sprint #7 los otros programadores comenzaron el módulo de Reportes, mientras tanto yo hice el módulo del Calendario, para éste utilicé el Scheduler de DevExtreme, con el cual me fue mucho más sencillo crear la funcionalidad que tenía que mostrar, esa funcionalidad fue:

1. Sincronizar el calendario del CRM con mi calendario de Google.
2. Mostrar las tareas que tenía asignadas en mi calendario.
3. Crear la funcionalidad para publicar mi calendario con un horario de trabajo definido dentro de una página, un correo o en cualquier sitio que pudiera embeber un script y que los usuarios pudieran crear citas y verlas reflejadas en mi calendario.

A su vez también creé una tab en mis 4 módulos principales a las que se les agregó el calendario y en éste mostraba todas las tareas asignadas de ese ítem en específico.

Al finalizar esto, los programadores también tenían completo el proyecto de Reportes, le creé su módulo en el menú y al presionar el botón del módulo mandamos abrirlo en una nueva pestaña, con esto navegando entre las pestañas podíamos interactuar tanto con el CRM como con el módulo de Reportes.

Con todo esto pude finalizar mi Sprint # 7 y prácticamente terminamos nuestro objetivo inicial, el cual era tener la misma funcionalidad o casi toda la del CRM anterior y mejorar nuestro CRM con unos cuantos módulos más y cosas que eran un valor agregado para el uso del cliente.

Posterior a ello y como en todo nuevo programa se le hicieron las pruebas de carga y estrés necesarias para probar el uso correcto del sistema. Se hicieron los manuales y presentaciones correspondientes para poder brindar una capacitación. Y comenzamos la liberación del sistema siguiendo un esquema de etapas de liberación: etapa alfa, etapa beta, etapa final.

## 9. Resultados obtenidos.

Hablando de resultados puedo comentar que nos funcionó bastante bien.

Lo primero que hicimos fue dentro de la empresa migrar del CRM anterior al nuevo, esto poniendo como ejemplo de que hay que utilizar lo hecho en casa.

Debo mencionar que al principio la curva de aprendizaje fue un poco lenta, los usuarios dentro de la empresa estaban reacios al cambio pese a que todo el sistema funcionaba de manera mucho más fácil que el anterior y dimos la capacitación necesaria para entender el uso correcto del CRM.

Una vez que la gente empezó a cooperar y a utilizar más el nuevo CRM, como en todo nuevo programa detectamos ciertos bugs y les dimos sus correcciones necesarias realizando mantenimiento cada cierto tiempo. Como ventaja teníamos a toda nuestra gente dentro de la empresa que eran usuarios finales como testers y esto nos ayudó para poder dejar una versión corregida en un tiempo mucho más corto.

Posteriormente algunos de los clientes con el CRM anterior decidieron migrar sus datos a este nuevo CRM por lo cual se realizó todo un proceso de migración de datos a esta nueva versión. La retroalimentación de ellos es que fue mucho más fácil el manejo del CRM y quedaron muy satisfechos ya que como prácticamente todas las páginas eran iguales, es decir contenían un grid principal les era mucho más fácil entender y manejar todo el CRM, sin embargo, al igual que la gente dentro de la empresa en la que trabajo a ellos también les fue un poco difícil la curva de aprendizaje y sobre todo el acostumbrarse a utilizarlo.

Otra cosa que hay que mencionar es que no todos los clientes se quisieron cambiar a esta nueva versión, por lo cual tuvimos que dejar el sitio del CRM viejo en línea. Hoy en día ese sitio aún funciona y tenemos una gran cantidad de clientes utilizando esa versión del CRM.

Para los nuevos clientes que llegaron ya no se les ofreció la versión anterior del CRM sino esta nueva y fue mucho más sencillo que se adaptaran al CRM que los que se migraron de la versión anterior. Tuvimos buena retroalimentación y como en todo también peticiones para agregar nuevos requerimientos.

## **10. Mejoras al sistema a futuro.**

Actualmente el CRM está trabajando de manera correcta y estable.

Hace mucho tiempo que no tenemos problemas de bugs y le seguimos dando el mantenimiento necesario cada determinado tiempo.

A su vez también hemos creado nueva funcionalidad, entre tantas puedo mencionar que agregamos funcionalidad de telefonía, esto mediante un teléfono SIP con extensiones para todos los usuarios, por lo cual hoy en día podemos marcarnos entre los usuarios de la licencia y a los clientes desde nuestras propias extensiones SIP.

Con el paso del tiempo y al aprendizaje que hemos tenido manejando y vendiendo CRM y obviamente debido al aumento de la competencia y a las necesidades tanto de la empresa como de nuestros clientes nos dimos cuenta de que hoy en día lo que las empresas quieren es tener sus propios procesos definidos, no unos predefinidos como en el caso del CRM o ERP o todos estos sistemas similares.

Por lo cual decidimos que el CRM como tal iba cambiar a una especie de Business Process Management, en el cual mediante un sistema de creación de formas el cliente puede crear sus propios procesos y que éstos funcionen a través de cierta cantidad de pasos definidos por el cliente.

Actualmente estamos trabajando en esa nueva versión del sistema, debo mencionar que he reutilizado mucho código de mi versión anterior por lo cual ha sido mucho más fácil el realizar este proyecto y estamos a escasos meses de liberarlo.

## **11. Conclusiones.**

Como conclusión a mi Informe de Trabajo Profesional, podemos decir que se logró cumplir el objetivo principal, el cual fue elaborar un CRM usando herramientas de programación actuales.

Logramos realizar de manera sencilla nuestro proyecto gracias a la metodología ágil SCRUM, con la cual una vez más pude comprobar que es una de las mejores metodologías para realizar un proyecto desde cero o bien darle seguimiento a uno ya creado debido a su fácil uso e implementación. Tengo que mencionar que a lo largo de mi vida laboral he probado diversas metodologías para la creación de un proyecto y ésta por mucho es una de mis favoritas por su simpleza.

Como producto hay que destacar que obtuvimos un CRM adaptado a las necesidades que se tenían tanto en la empresa como de parte de nuestros clientes, todo esto siguiendo todos los protocolos de calidad y seguridad para que nuestro CRM pudiera ser comprado por empresas que requieren altos estándares, en especial por los bancos.

Hoy en día el CRM está muy estable y seguirá vendiéndose. Actualmente estamos realizando un cambio de concepto de un CRM a un BPM es muy probable que la versión de CRM se quede como está ahorita y se venda a la par junto con la nueva versión de BPM, eso aún no se decide y va a depender del área comercial si se mantiene o no.

Antes de concluir quiero agradecer a la empresa en la que actualmente laboré por haberme otorgado el permiso para poder presentar este informe.

Y por último un gran agradecimiento al Ing. Alberto Templos Carbajal por haberme apoyado aceptando ser mi asesor en este proceso.

## 12. Bibliografía.

ANDER, E. Buscando la sinergia en Trabajo Social: Técnicas de Reuniones de Trabajo. Bogotá: Colatina, 1982

BOSE, R., Customer Relationship Management: Key components for IT success, 2002

LARMAN, Craig Applying UML and Patterns New Jersey, U.S.A Prentice Hall, 2002

LIMA, B. Contribución a la Metodología de Trabajo Social. Venezuela: Universidad Central de Venezuela, 1963.

SOMMERVILLE, Ian Software Engineering 7th Edition Reading, Massachussets, U.S.A. Addison Wesley, 2004

JIRA, Herramientas de gestión ágil de proyectos para equipos de software (atlassian.com), Recuperado el día 24 de mayo de 2021, <https://www.atlassian.com/es/software/jira/agile#:~:text=Jira%20Software%20es%20una%20herramienta,kanban%20o%20la%20tuya%20propia>

CONFLUENCE, Confluence | Tu espacio de trabajo remoto en equipo (atlassian.com), Recuperado el día 24 de mayo de 2021, <https://www.atlassian.com/es/software/confluence>

DEVEXPRESS, .NET UI Controls for Developers of Mobile, Desktop, Web, Reporting & BI Apps (devexpress.com), Recuperado el día 24 de mayo de 2021, <https://www.devexpress.com/>

DEVEXTREME, DevExtreme – Javascript UI Components for Angular, React, Vue and jQuery by DevExpress, Recuperado el día 24 de mayo de 2021, <https://js.devexpress.com/>

VISUAL STUDIO, IDE de Visual Studio, editor de código, Azure DevOps y App Center – Visual Studio (microsoft.com), Recuperado el día 24 de mayo de 2021, <https://visualstudio.microsoft.com/es/>

MAILCHIMP, Plataforma Integrada de marketing todo en uno para pequeñas empresas | MailChimp, Recuperado el día 24 de mayo de 2021, <https://mailchimp.com/es/>

SQL SERVER, Descargas de SQL Server | Microsoft, Recuperado el día 24 de mayo de 2021, <https://www.microsoft.com/es-mx/sql-server/sql-server-downloads>

UNIVERSITAT OBERTA DE CATALUNYA (UOC), Gestión de Proyectos Informáticos Metodología SCRUM, Recuperado el día 24 de mayo de 2021,

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

PROYECTOS AGILES.ORG, Que es SCRUM – Proyectos Agiles (proyectosagiles.org), Recuperado el día 24 de mayo de 2021, <https://proyectosagiles.org/que-es-scrum/>