



Tarea 4: Implementación OO

Symdex

Cliente: Symbiose

Empresa: Blastosoft

Integrantes: Juan Arredondo

Flavio Ramos

Alberto Rodríguez

Matías Tobar

18 de Julio de 2019

1. Resumen ejecutivo

Symdex es una aplicación web donde se permiten dos usuarios. El administrador, el cual la aplicación le permite crear propuestas, editarlas, buscarlas y eliminarlas, como también asignar propuestas a un cliente, junto con poder ver las estadísticas de cierto cliente. Por ende, el objetivo general de Symdex para el usuario administrador es gestionar las propuestas que se le asignan a cada cliente, además ofrece la posibilidad de la indexación de estas.

Para el otro usuario de la aplicación el cual se le asigna el nombre de “usuario”, es un trabajador de la empresa Symbiose, al cual se le asigna solamente la posibilidad de ver las propuestas técnicas de la aplicación, no tiene la posibilidad de gestionar nada, ni tampoco ver las propuestas económicas.

El siguiente escrito busca abarcar el desarrollo de la fase de Diseño OO de OMT++/UML y los resultados obtenidos de esta son el programa ejecutable, con los manuales de instalación y de usuario asociados.

Tabla de contenidos

| | |
|--|-----------|
| 1. Resumen ejecutivo | 2 |
| 2. Introducción | 1 |
| 3. Plataformas y herramientas de desarrollo utilizadas | 3 |
| 4. Consideraciones de implementación del diseño | 4 |
| 5. Requisitos de instalación | 9 |
| 6. Procedimiento de instalación | 10 |
| 7. Manual de usuario | 12 |
| 7.1. Funcionalidades de sistema | 12 |
| 7.1.1. Agregar una propuesta y ver lista de propuestas | 12 |
| 7.1.2. Agregar un cliente y ver lista de clientes | 13 |
| 7.1.3. Agregar un usuario y ver lista de usuarios | 14 |
| 8. Resultados del proyecto | 15 |
| 9. Conclusiones | 17 |
| 10. ANEXO | 18 |
| 10.1. Código Fuente | 18 |

Índice de figuras

| | | |
|-----|--|----|
| 1. | Caso de Uso Ingresar a la plataforma. | 4 |
| 2. | Caso de Uso Listar propuestas. | 5 |
| 3. | Diagrama de clases. | 7 |
| 4. | Diagrama de despliegue o procesos. | 8 |
| 5. | Clonación de los repositorios. | 10 |
| 6. | Ejecución de npm install en el repositorio de la capa de presentación. | 10 |
| 7. | Ejecución de npm en la carpeta de la capa de presentación. | 11 |
| 8. | Ejecución de npm run serve en la carpeta de la capa de presentación. | 11 |
| 9. | Ejecución de npm run serve en la carpeta de la capa de presentación. | 11 |
| 10. | Funcionalidad: Agregar Propuesta. | 12 |
| 11. | Funcionalidad: Listado de Propuestas. | 12 |
| 12. | Funcionalidad: Agregar Cliente. | 13 |
| 13. | Funcionalidad: Listado de Clientes. | 13 |
| 14. | Funcionalidad: Agregar Usuario. | 14 |
| 15. | Funcionalidad: Listado de Usuarios. | 14 |

2. Introducción

En el contexto del curso Fundamentos de Ingeniería de Software impartido por el Departamento de Ingeniería Informática de la Universidad de Santiago de Chile se ha asignado a los estudiantes la realización de un proyecto informático en colaboración con la empresa Symbiose a modo de llevar a la práctica el conocimiento adquirido en las clases teóricas del curso.

La empresa está actualmente en un proceso de expansión empresarial. Su manera de trabajo es a través de propuestas, las cuales se redactan y envían a potenciales clientes, en base a una idea planteada por la empresa, de 50 propuestas enviadas, 1 de ellas, por lo general, es aceptada. La forma de manejar el nivel de propuestas es difícil con esta herramienta, ya que no brinda las facilidades que se necesitan para la gestión de las propuestas. Por lo que el cliente (Symbiose) necesita una nueva herramienta que les permita tener control de las propuestas (a qué cliente se ha contactado y a qué cliente se le ha realizado seguimiento).

La solución que propone el grupo Symdex consiste en crear una aplicación web que ayude y presente mejoras para un proceso ya existente, en este caso el manejo de las propuestas técnicas y económicas que la empresa Symbiose envía a sus clientes. El sistema actual, que ordena las propuestas a través de hojas de cálculo, limita la cantidad de información que se puede manejar sin que el proceso llegue a ser lento y engorroso, lo cual es indeseado para una empresa la cual desea expandir su red de clientes y sus posibilidades de obtener proyectos.

En palabras simples trata de un sistema de almacenamiento que permita mantener organizadas las propuestas realizadas por la compañía, como también brindar otra información de importancia para el administrador de la plataforma como lo pueden ser los equipos de trabajo destinados a cada proyecto, la ganancia que le produce, los plazos de los contratos o información relacionada a la empresa cliente. El sistema también ha de permitir el ingreso de los desarrolladores pertenecientes a la empresa y darles la opción de obtener las propuestas técnicas realizadas.

En primera instancia, las directrices del proyecto fueron adquiridas por el alumnado por medio de una presentación realizada por los representantes de Symbiose (el cliente) en la que los estudiantes pudieron escuchar y documentar las ideas dadas por el cliente, así como

también aclarar algunos puntos y resolver inquietudes que iban surgiendo conforme avanza la presentación.

De manera posterior a las primeras impresiones adquiridas de la entrevista anteriormente mencionada, los estudiantes han avanzado en el proyecto realizando tanto presentaciones orales como informes escritos, siguiendo la metodología OMT++/UML, siendo las tres anteriores a esta, la captura de requisitos, el análisis orientado a objetos y el diseño orientado a objetos.

Para la presente entrega, que consiste en la programación orientada a objetos, se pretende plasmar de la manera más consistente posible todo el trabajo realizado hasta este punto y con ello poder entregar una solución informática de la mayor calidad y utilidad posible para el cliente.

3. Plataformas y herramientas de desarrollo utilizadas

Para el proceso de desarrollo del proyecto, las plataformas y herramientas utilizadas fueron:

- **GitHub:** Plataforma de desarrollo colaborativo para almacenar proyectos utilizando el sistema de control de versiones Git, donde se encuentran los dos repositorios del proyecto, uno para lo que es la parte de modelos y controladores o back-end y un segundo repositorio donde se encuentra la capa de presentación del proyecto o front-end.
- **SublimeText, VisualCode:** Todos estos editores de texto y de código fuente utilizados para visualizar de mejor forma el código programado por el equipo de trabajo.
- **IntelliJ IDEA:** Un entorno de desarrollo integrado o IDE, para el desarrollo de programas informáticos en el lenguaje de programación Java. Utilizado por el equipo de desarrollo para facilitar la compilación y ejecución del programa.
- **MongoDB:** Motor de base de datos no relacionales orientado a documentos y cuyo uso se debe a una solicitud explícita del cliente.
- **Vue.js:** es un marco de JavaScript de código abierto para crear interfaces de usuario y aplicaciones.
- **Postman:** Cliente HTTP que permite probar servicios web, utilizado para la comprobación de las operaciones CRUD de la información dentro de la base de datos.
- **Bootstrap:** es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.
- **Spring:** Es un framework para el desarrollo de aplicaciones de código abierto para la Plataforma Java, es la herramienta que permite la conexión entre lo que son las capas de modelo y controladoras del 10 proyecto y lo que es la capa de presentación. Su uso en este proyecto fue por solicitud del cliente.

4. Consideraciones de implementación del diseño

Aunque los diagramas arquitecturales no fueron directamente rediseñados en la etapa de implementación, los diagramas finales carecen de ciertos elementos debido a que no pudieron llegar a ser implementados por completo. Para lo que fue la implementación de la solución se tomaron en cuenta dos casos de uso principales que fueron los implementados.

| Ingresar Propuesta | |
|--------------------|--|
| Actor(es) | Administradores |
| Precondición | Ingresar a la plataforma y estar en la vista principal |
| Descripción | Actor clickea el boton de agregar propuesta que se encontrará al costado izquierdo de la vista principal, el sistema cambiará de vista. |
| | En la nueva pestaña ingresará el título de la propuesta junto con una breve descripción de la misma, mientras el sistema mostrará lo ingresado por el actor por pantalla en la aplicación. |
| | Se clickea el botón para guardar los cambios y el sistema mostrará por pantalla una nueva vista confirmando el ingreso a sistema de la nueva propuesta |
| Postcondición | La propuesta se encuentra ingresada al sistema y este además mostrará dos nuevos botones para cambiar de vista. |
| Excepción | (paso 3) Carencia de alguno de los datos pedidos, generará una propuesta con carencia de datos y por ende una propuesta la cual no se podrá manipular según el sistema |

Figura 1: Caso de Uso Ingresar a la plataforma.

| | |
|-------------------|--|
| Listar Propuestas | |
| Actor(es) | Administradores, usuarios |
| Precondición | Ingresar a la plataforma y encontrarse en vista principal, además de que en el repositorio se encuentre con algún dato. |
| Descripción | 1. Haciendo click en el boton de listado, el sistema procederá a cambiar la vista de la página. |
| | 2. Una vez cambiada la vista el sistema mostrará el listado con las propuestas que encuentre en la base de datos. |
| | |
| Postcondición | El sistema mostrará el listado de propuestas por pantalla |
| Excepción | (paso 1) Problemas con la conexión con el servidor para acceder al contenido del listado, imposibilitando la vista del listado a través de la aplicación.. |

Figura 2: Caso de Uso Listar propuestas.

Al momento de implementar la solución se consideró solo un controlador el cual contiene las rutas que conectan la vista con los modelos, si bien se había planeado tener un controlador por cada vista se prefirió tener solo uno que manejara todo el flujo de información entre las capas del MVC.

No se cuenta con una vista que permita restringir ciertas operaciones, es por esto que todas las operaciones realizadas se consideraban hechas por un usuario de tipo administrador. Finalmente el diagrama de clases de lo implementado se puede ver en la Figura 3.

Por otra parte el diagrama de despliegue también sufrió ligeros cambios debido al conocimiento adquirido sobre protocolos y comunicación entre componentes.

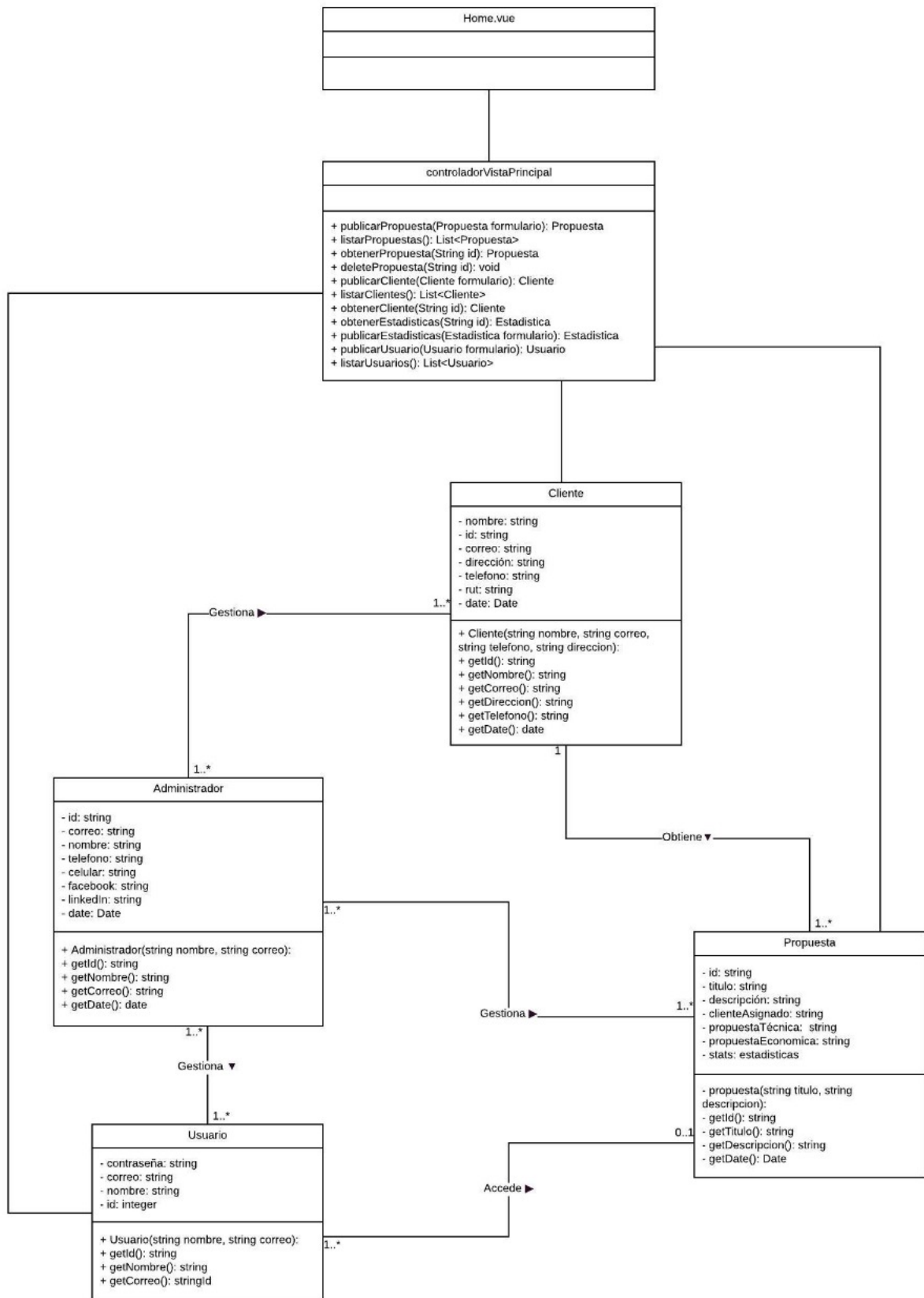


Figura 3: Diagrama de clases.

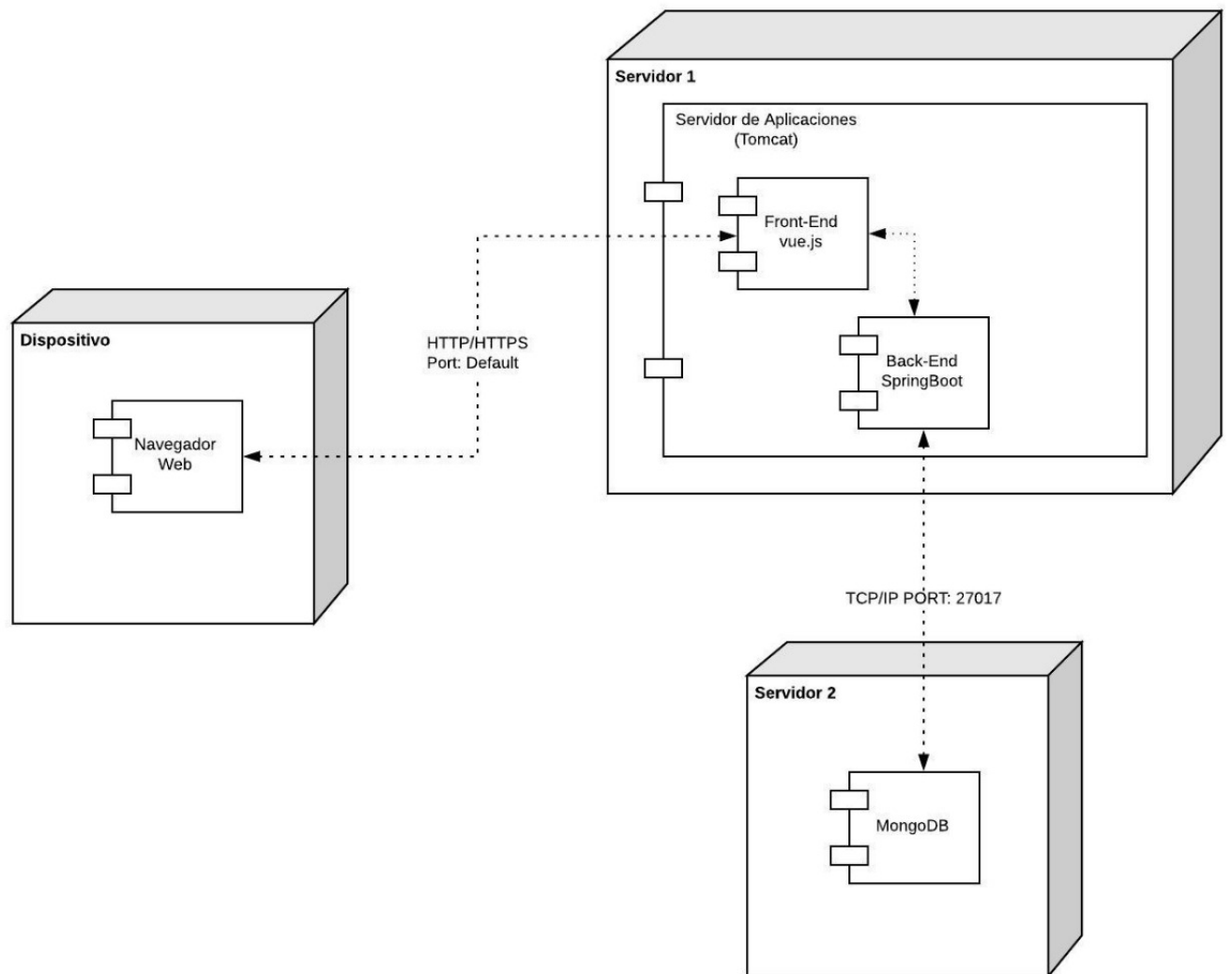


Figura 4: Diagrama de despliegue o procesos.

5. Requisitos de instalación

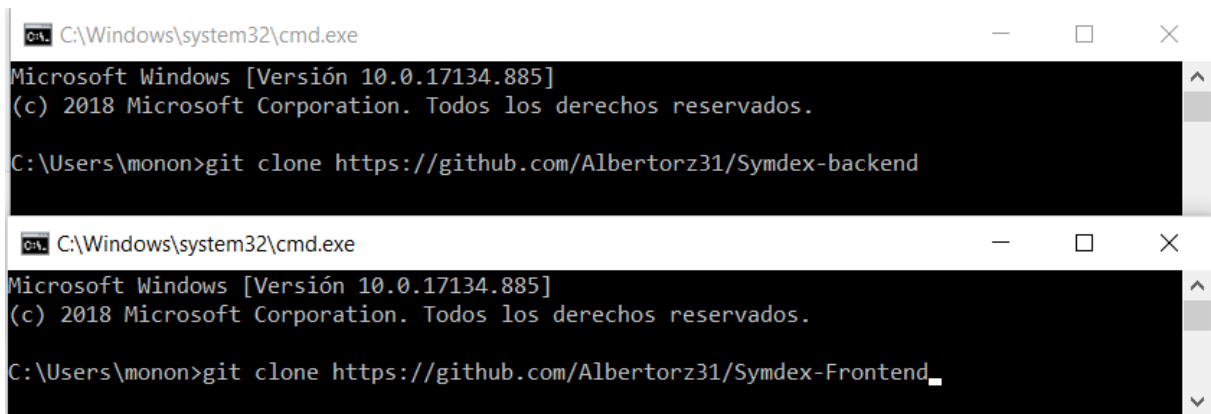
Para poder realizar la instalación y posterior ejecución del programa implementado, se requieren algunas herramientas y sus respectivas variables de entorno instaladas en el equipo. Estas son:

- **Git:** Permite la descarga del proyecto desde GitHub.
- **Vue js:** Permite la compilación y ejecución de la capa de presentación de la aplicación.
- **Maven:** Permite la compilación y ejecución de la capa controladora y de modelos de la aplicación.
- De otra manera, es posible realizar la compilación de ambos mediante **IntelliJ IDEA**, ya que esta IDE facilita el trabajo de detección e instalación automática de dependencias dentro de un proyecto.

6. Procedimiento de instalación

Para instalar y ejecutar Symdex se debe proceder de la siguiente forma, en caso de hacerse por consola:

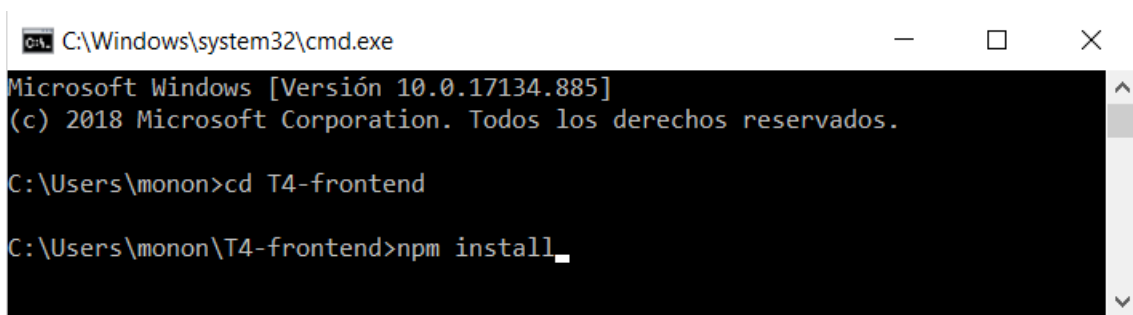
1. Abrir la consola en la carpeta donde se desea guardar el programa, una vez hecho esto, clonar los repositorios <https://github.com/Albertorz31/Symdex-backend> y <https://github.com/Albertorz31/Symdex-Frontend> por medio de la ejecución del comando “git clone + la respectiva URL” (ver Figura 5).



The figure consists of two screenshots of a Windows command prompt window. The title bar of the window is 'C:\Windows\system32\cmd.exe'. The first screenshot shows the command 'git clone https://github.com/Albertorz31/Symdex-backend' being entered at the prompt 'C:\Users\monon>'. The second screenshot shows the command 'git clone https://github.com/Albertorz31/Symdex-Frontend' being entered at the prompt 'C:\Users\monon>'. Both screenshots show the standard Windows copyright notice: 'Microsoft Windows [Versión 10.0.17134.885] (c) 2018 Microsoft Corporation. Todos los derechos reservados.'

Figura 5: Clonación de los repositorios.

2. Para el software de la capa de presentación ubicado en la carpeta “T4-frontend”, ejecutar el comando “npm install” cerrar la consola (ver Figura 6).



The figure is a screenshot of a Windows command prompt window. The title bar is 'C:\Windows\system32\cmd.exe'. The prompt shows the user has navigated to the 'T4-frontend' directory with the command 'cd T4-frontend'. The command 'npm install' is being entered at the prompt 'C:\Users\monon\T4-frontend>'. The standard Windows copyright notice is also visible.

Figura 6: Ejecución de npm install en el repositorio de la capa de presentación.

3. En la carpeta “T4-backend”, donde se ubica las capas de modelos y controladores, abrir la consola y ejecutar el comando “mvn springboot:run” (ver Figura 7).

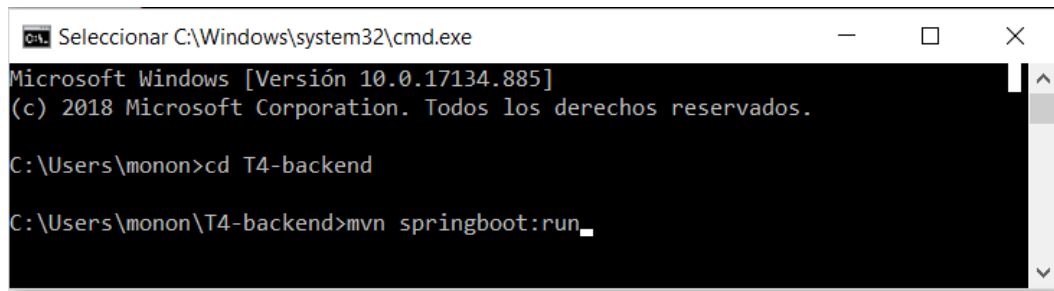


Figura 7: Ejecución de npm en la carpeta de la capa de presentación.

4. Posteriormente en la misma carpeta del paso 2, abrir una nueva de la consola y ejecutar el comando “npm run serve”(ver Figura 8).

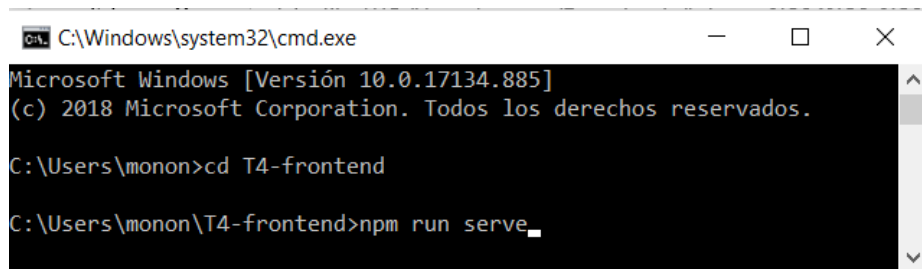


Figura 8: Ejecución de npm run serve en la carpeta de la capa de presentación. .

5. Una vez hecho esto, se abrirá una nueva pestaña en el navegador web y mostrará la ventana principal de la plataforma y el usuario puede ya utilizarla (ver Figura 9).

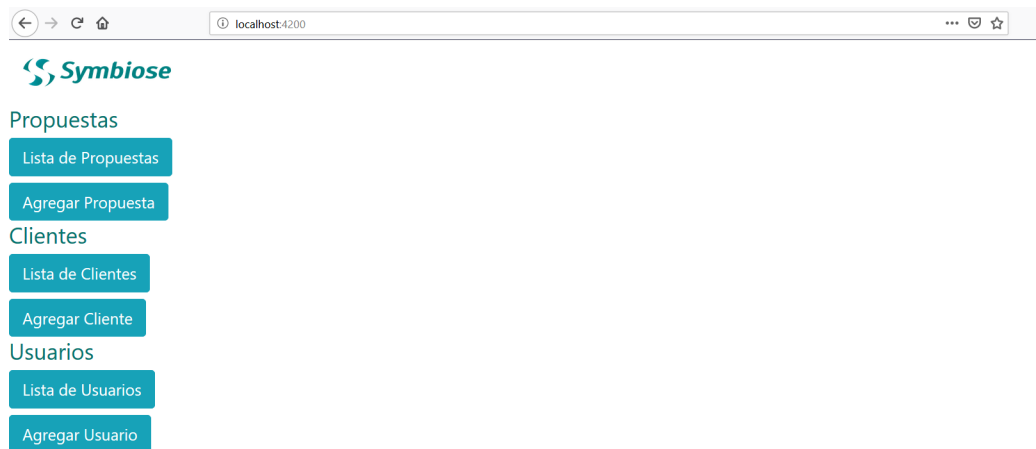


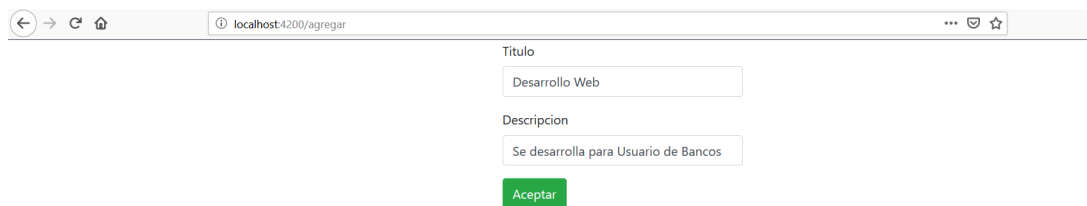
Figura 9: Ejecución de npm run serve en la carpeta de la capa de presentación.

7. Manual de usuario

7.1. Funcionalidades de sistema

7.1.1. Agregar una propuesta y ver lista de propuestas

Para publicar una propuesta, el administrador escoge el botón de “Agregar Propuesta”, en donde le aparecerá otra pestaña la cual indica que debe llenar los campos “Título” “Descripción” de la propuesta, después aparecerá un mensaje de que la propuesta se ha agregado correctamente, esta es agregada en la base de datos y puede ser observada presionando el botón “Lista de Propuestas” donde ahí se verá agregada. (ver Figura 10).



Titulo

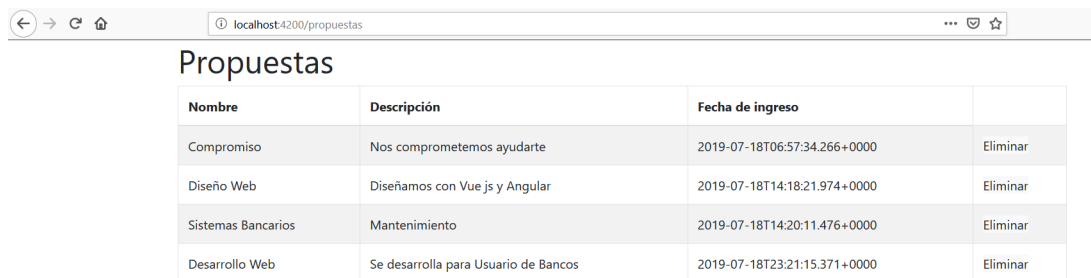
Desarrollo Web

Descripcion

Se desarrolla para Usuario de Bancos

Aceptar

Figura 10: Funcionalidad: Agregar Propuesta.

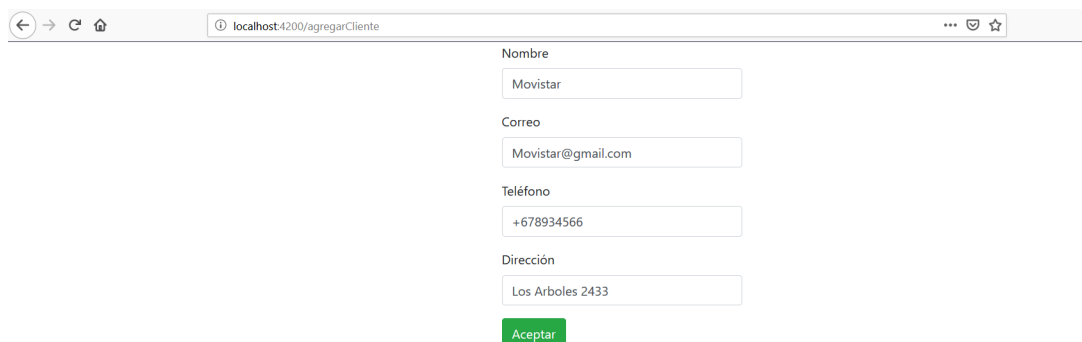


| Nombre | Descripción | Fecha de ingreso | |
|--------------------|--------------------------------------|------------------------------|----------|
| Compromiso | Nos comprometemos ayudarte | 2019-07-18T06:57:34.266+0000 | Eliminar |
| Diseño Web | Diseñamos con Vue.js y Angular | 2019-07-18T14:18:21.974+0000 | Eliminar |
| Sistemas Bancarios | Mantenimiento | 2019-07-18T14:20:11.476+0000 | Eliminar |
| Desarrollo Web | Se desarrolla para Usuario de Bancos | 2019-07-18T23:21:15.371+0000 | Eliminar |

Figura 11: Funcionalidad: Listado de Propuestas.

7.1.2. Agregar un cliente y ver lista de clientes

Para agregar un cliente, el administrador escoge el botón de “Agregar Cliente”, en donde le aparecerá otra pestaña la cual indica que debe llenar los campos “Nombre”, “Correo”, “Teléfono” “Dirección” de la propuesta, después aparecerá un mensaje de que el cliente se ha agregado correctamente, esta es agregada en la base de datos y puede ser observada presionando el botón “Lista de Clientes” donde ahí se verá agregada. (ver Figura 12).



Nombre
Movistar

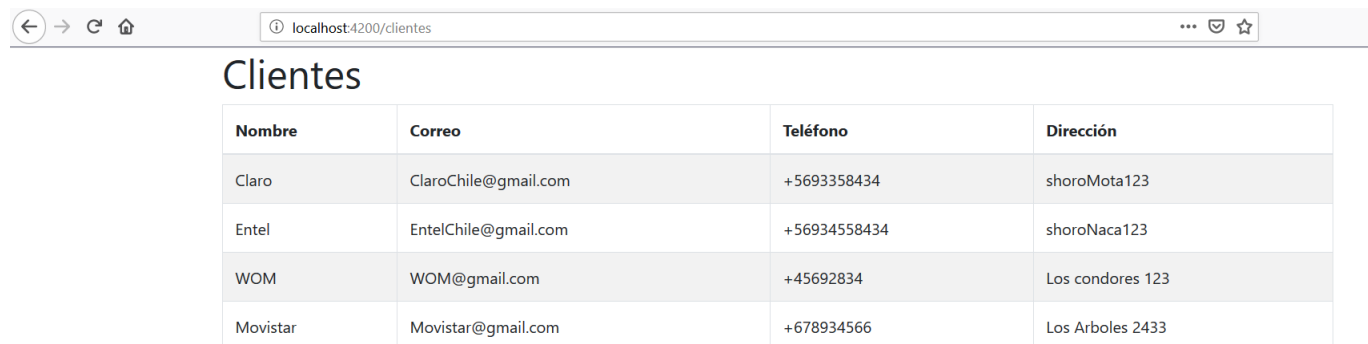
Correo
Movistar@gmail.com

Teléfono
+678934566

Dirección
Los Arboles 2433

Aceptar

Figura 12: Funcionalidad: Agregar Cliente.

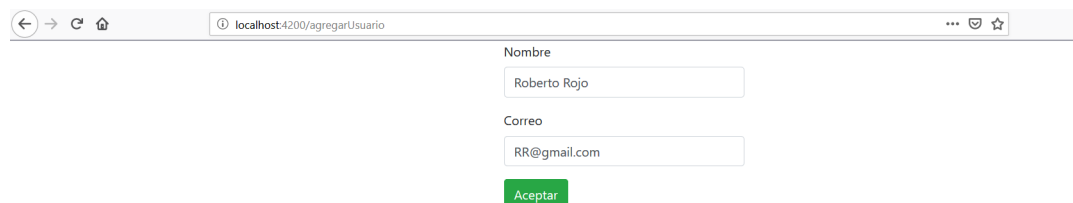


| Nombre | Correo | Teléfono | Dirección |
|----------|----------------------|--------------|------------------|
| Claro | ClaroChile@gmail.com | +5693358434 | shoroMota123 |
| Entel | EntelChile@gmail.com | +56934558434 | shoroNaca123 |
| WOM | WOM@gmail.com | +45692834 | Los condores 123 |
| Movistar | Movistar@gmail.com | +678934566 | Los Arboles 2433 |

Figura 13: Funcionalidad: Listado de Clientes.

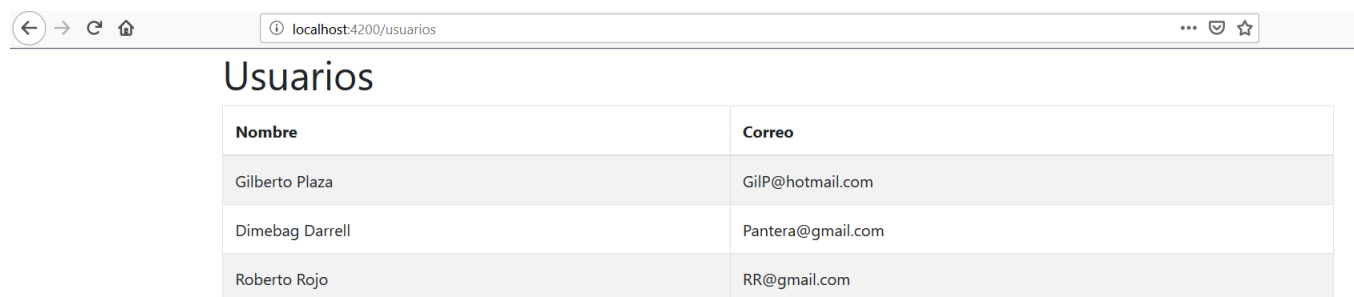
7.1.3. Agregar un usuario y ver lista de usuarios

Para agregar un cliente, el administrador escoge el botón de “Agregar Usuario”, en donde le aparecerá otra pestaña la cual indica que debe llenar los campos “Nombre” “Correo” de la propuesta, después aparecerá un mensaje de que el usuario se ha agregado correctamente, esta es agregada en la base de datos y puede ser observada presionando el botón “Lista de Usuarios” donde ahí se verá agregada. (ver Figura 14).



A screenshot of a web browser window showing a form titled "Agregar Usuario". The browser's address bar displays "localhost:4200/agregarUsuario". The form contains two input fields: "Nombre" with the text "Roberto Rojo" and "Correo" with the text "RR@gmail.com". Below these fields is a green button labeled "Aceptar".

Figura 14: Funcionalidad: Agregar Usuario.



A screenshot of a web browser window showing a table titled "Usuarios". The browser's address bar displays "localhost:4200/usuarios". The table has two columns: "Nombre" and "Correo". It contains three rows of user data.

| Nombre | Correo |
|-----------------|-------------------|
| Gilberto Plaza | GilP@hotmail.com |
| Dimebag Darrell | Pantera@gmail.com |
| Roberto Rojo | RR@gmail.com |

Figura 15: Funcionalidad: Listado de Usuarios.

8. Resultados del proyecto

En vista de todo el trabajo realizado, como resultado de esta cuarta entrega, correspondiente a la implementación orientada a objetos se ha obtenido una aplicación web completamente funcional hecha en base a algunos requerimientos y casos de uso de la primera entrega, siguiendo el análisis y el diseño orientado a objetos definidos en las entregas dos y tres, respectivamente.

1. **Cantidad de diálogos (o vistas) implementados:** La aplicación cuenta con 6 vistas que constituyen las funciones de agregación y las listas.
2. **Cantidad de clases clasificadas según la arquitectura del sistema (MVC++ o MVC):** Symdex tiene un total de cinco clases según la arquitectura MVC++, estas son Propuesta, Cliente, Usuario, Estadísticas y Administrador. Estas no fueron utilizadas en el frontend, pero si se encuentran en el backend del proyecto.
3. **Número de componentes de desarrollo debidamente identificados:** Se tienen 3 componentes principales, que son Módulo Propuesta, Módulo Cliente y Módulo Usuario.
4. **Número de componentes de proceso debidamente identificados:** Se tienen 4, que son Cliente, Front-End, Back-End y Database.
5. **Tamaño en bytes de cada componente de desarrollo:** El tamaño asociado al archivo que contiene el Front-End es de 110 MB, atribuimos este tamaño a la cantidad de librerías y elementos visuales que requiere, mientras que por parte del Back-End tiene un peso de 185 KB, al simplemente documentos de texto con código su tamaño es mucho más reducido.
6. **Tamaño de cada ejecutable:** el tamaño de cada ejecutable es aproximadamente 50 kb
7. **Tiempo dedicado al desarrollo del proyecto:**
 - **Entrega 1:** 16 hrs aproximadamente, dividida en tres reuniones presenciales, de aproximadamente 3 horas cada una, 3 horas para la redacción colaborativa del informe, 2 horas para la confección colaborativa de la presentación, y 2

- **Entrega 2:** 20 hrs, divididas en tres reuniones presenciales, de aproximadamente 2 horas cada una, 6 horas para la redacción colaborativa del informe, 4 horas para la confección de diagramas y prototipos de vistas, 2 horas para la confección colaborativa de la presentación, y 2 horas para la preparación y coordinación presencial de la presentación.
- **Entrega 3:** 23 hrs, divididas en tres reuniones presenciales, de aproximadamente 3 horas cada una, 6 horas para la redacción colaborativa del informe, 6 horas para la confección y modificación de diagramas, 1 hora para la confección colaborativa de la presentación, y 2 horas para la preparación y coordinación presencial de la presentación.
- **Entrega 4:** 20 hrs, lamentablemente para la última entrega, no hubo reuniones presenciales debido al exceso de cosas y la falta de tiempo, esto se ve reflejado en la aplicación que es bastante pobre. las 20 hrs son de prácticamente de aprender las tecnologías y programación extrema.

9. Conclusiones

A lo largo del desarrollo de este proyecto el cual tenía como motivo ser un “incremento”, se nos presentaron muchas herramientas para apoyarnos en ese objetivo, una de las principales en la cual nos basamos fue OMT++ (Object Modeling Technique), una metodología orientada a objetos, a través de la cual se nos disponen distintos diagramas, en los cuales se puede ver el cómo se llevará a cabo el proyecto, junto con distintos símbolos para entender esos modelos, esto facilita el entendimiento del proyecto para aquellos que conociesen también el modelado, ya fuese para el Product Owner o el equipo de desarrollo. A través de este modelado de objetos logramos obtener tanto un nivel de abstracción mayor, como una nueva manera de poder representar aquella idea que representaría el proyecto, junto con el hecho de acercarnos un poco más al ámbito empresarial que se tiene en la industria del desarrollo de software, el cual en nuestro caso usamos para el desarrollo web, para una aplicación.

Si bien el proyecto no se desenvolvió de la manera presupuestada por parte del equipo, el desarrollo y la presentación de este último, dejó enseñanzas y experiencia las cuales servirán a futuro para poder tener un respaldo empírico de lo que se debe hacer en este tipo de casos, por ejemplo, al tratar con “nuevas tecnologías” para los desarrolladores, se espera que, se encuentre respaldo en documentos y en caso de que esto no sea suficiente, reevaluar el desarrollo del proyecto acotando las posibles funciones “previstas” para ser entregadas en los incrementos, así como el hecho de, si se está en una situación de tiempo acotado, en el cual posiblemente no se desarrollen las funciones pedidas por el cliente, en post de tener una “base de funciones” se esperará que se desarrollen ciertas funciones y que esas funciones se encuentren en un ámbito buen ámbito funcional, llámese a esto último como, con Robustez y Escalabilidad, además de considerar la buena implementación de estas funciones (“...¿cuántos clicks requiero para usarla?...”).

10. ANEXO

10.1. Código Fuente

El código fuente de la aplicación se encuentra separado en dos repositorios almacenados en la plataforma de desarrollo colaborativo GitHub. Los repositorios son privados, así que se necesita solicitar permiso para clonarlos. Los enlaces se listan a continuación, en el primero se encuentran las capas de modelo y controladores, mientras que en el segundo se encuentra la capa de presentación de la aplicación:

- <https://github.com/Albertorz31/Symdex-backend>
- <https://github.com/Albertorz31/Symdex-Frontend>