

Desarrollo de un software para el seguimiento de las aplicaciones utilizadas en los puertos de un computador utilizando tecnologías de libre distribución.

Resumen: Las empresas actualmente cuentan con la ayuda de las tecnologías de la información para el cumplimiento de sus actividades. Es por esto, que el control de la infraestructura tecnológica se ha vuelto de gran importancia para mitigar problemas de seguridad y tráfico pesado en la red. Este documento presenta una aplicación de escritorio para el seguimiento de las aplicaciones en uso dirigido a empresas que cuenten con una amplia infraestructura tecnológica y requieran hacer seguimiento del uso que se le da a sus equipos por medio de un informe generado por la aplicación. La aplicación se desarrolló con base en el lenguaje de modelado unificado y el uso de tecnologías de libre distribución. La aplicación cuenta con acceso de usuarios protegido por contraseña para asegurar la información que allí se encuentra y sus respectivas políticas de seguridad para un uso adecuado en la base de datos.

Palabras clave: Aplicación, Seguridad, TI, UML, OpenSource, puertos, infraestructura tecnológica.

1. INTRODUCCION

Los desarrollos y avances en tecnologías de la información aumentan rápidamente y con esto más usuarios en internet, más necesidades, más contextos a los que llegar, entre otras. Es por esto, que además de requerir avances también se necesita más control de la cantidad de usuarios y equipos que interactúan en cada infraestructura tecnológica.

La infraestructura tecnológica es “el servicio que ofrece el conjunto de dispositivos y aplicaciones necesarios para una empresa”. (Saavedra, 2018) Las empresas que ofrecen servicios y/o productos en TI se enfrentan a diversos desafíos para prestar un servicio con calidad confiable.

En todos los sectores la mayoría usa las TI como herramienta vital de trabajo. Uno de los principales problemas es el control de su infraestructura tecnológica. Esto, porque no controlarlo implica problemas de seguridad, tráfico pesado en la red. (Carvajal, Gómez, & Albarracín, 2020)

Este es el caso de la empresa Betulia S.A., que cuenta con numerosos equipos conectados lo que implica un tráfico amplio de dispositivos en la red. En el orden de ideas expresado la empresa requiere una solución tecnológica para hacer un control de su infraestructura que le permita conocer el uso que se da a los equipos conectados en la red de la empresa.

Este artículo presenta una aplicación que pretende satisfacer la necesidad mencionada anteriormente y detallar el proceso que se llevó a cabo en el desarrollo de esta. El lenguaje de modelado unificado fue la herramienta principal para el desarrollo de la aplicación ya que permite modelar la situación y la complejidad del proyecto. Adicionalmente se usó el correspondiente modelado de la base de datos para cumplir con las políticas de seguridad.

2. MARCO DE REFERENCIA

El marco conceptual que soporta la solución tecnológica presentada se compone de temas como modelo cliente-servidor, software open source (OSS), Sistemas gestores de bases de datos, UML y puertos de aplicaciones.

2.1. Modelo cliente-servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. (Schiaffarino, 2019)

2.2. Software Open Source

El software open source es un código diseñado de manera que sea accesible al público: todos pueden ver, modificar y distribuir el código de la forma que consideren conveniente. El software open source se desarrolla de manera descentralizada y colaborativa, así que depende de la revisión entre compañeros y la producción de la comunidad. (RedHat, 2020)

2.3. Sistemas gestores de bases de datos.

Un sistema gestor de bases de datos se define como como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos o BD. (Raffino, 2020)

2.4. UML

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. (Krall, 2018)

2.5. Puertos de aplicaciones.

Los puertos de aplicaciones se basan en puertos virtuales como puntos de conexión para el intercambio de información y la transmisión de datos. Un puerto suele estar numerado para de esta forma poder identificar la aplicación que lo usa. (Puntero Méndez, 2019)

3. METODOLOGIA

Para este proyecto se seguirá el modelo en cascada modificada. El modelo en cascada modificada es un proceso de desarrollo secuencial iterativo, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra y al ser modificada se permite que en cada etapa se haga realimentación de ellas. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo y de la misma manera de abajo hacia arriba si el proyecto se necesita cambios en la anterior etapa. (Domínguez, 2020)

A continuación, se explicará lo realizado en cada una de las etapas tratadas en la metodología

- **Análisis de problema:** en esta etapa se investiga lo necesario para la comprensión del problema se analizan causas y se plantean soluciones adecuadas para el proyecto.
- **Requerimientos:** análisis y alcance de la aplicación, definición de los requerimientos que el proyecto precisa para un buen resultado, los requerimientos aquí propuestos se modificaron por el alcance del proyecto.
- **Diseño:** en esta etapa se diseñan los diagramas de clases, casos de uso, actividades, entidad relación y también las interfaces visuales del usuario. Estos con el fin de dar forma, organizar y analizar cómo se comporta la aplicación, estos diseños se vieron afectados por modificaciones en el proceso.

- Implementación: en esta etapa se plasman los diseños en código y se hacen iteraciones entre esta fase y la fase de diseño por los cambios que se presentan en el proceso.
- Verificación: en esta etapa se realizan pruebas para intentar encontrar algún error o fallo en la aplicación.
- Correcciones: en esta etapa se corrigen errores encontrados en la etapa anterior y se itera constantemente con esta hasta mitigar la mayoría de los errores.

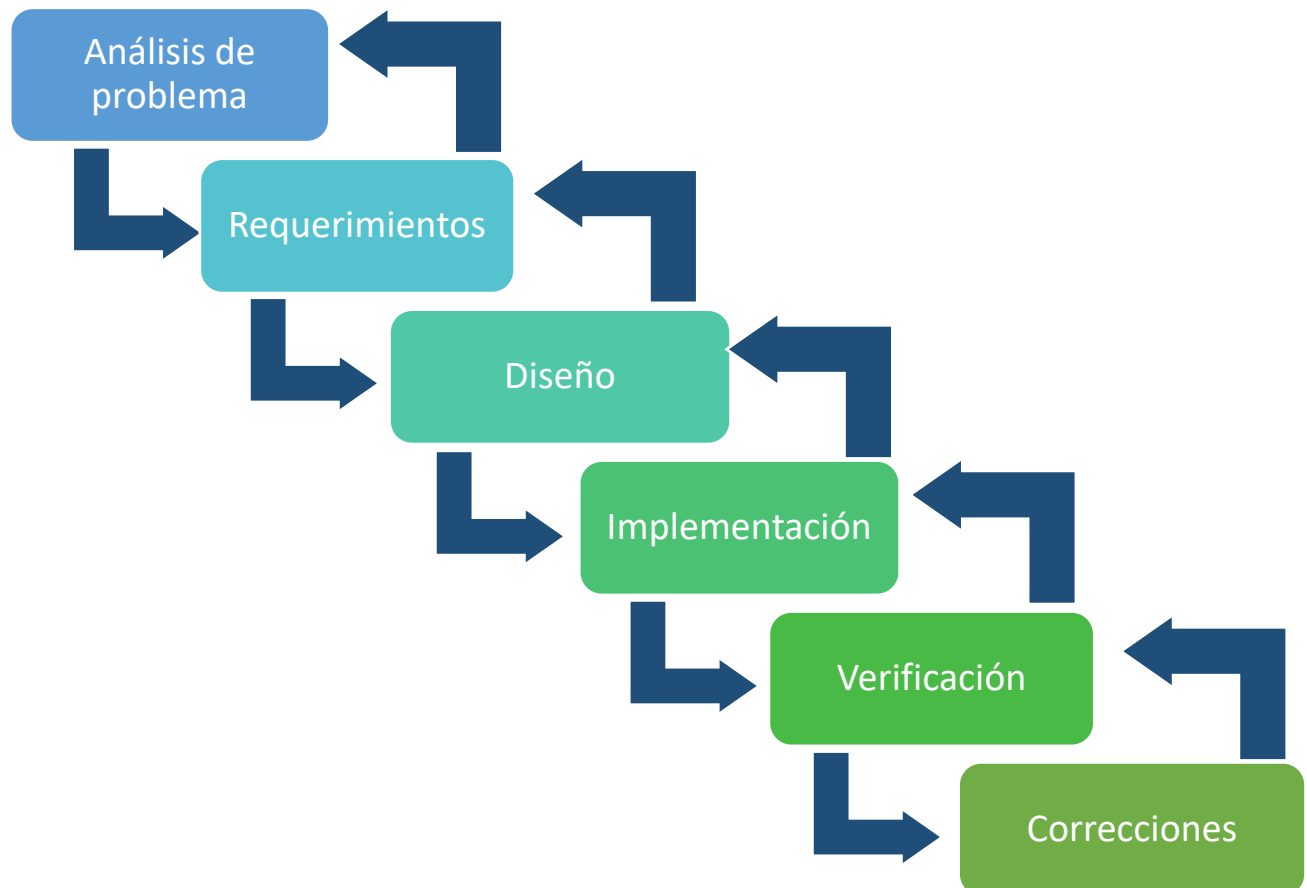


Figura 1. Representación gráfica de la metodología y sus etapas.

4. RESULTADOS

Los resultados del proyecto serán evidenciados en el orden de realización, el cual inicia con el análisis del problema en donde se estudiaron las causas del problema, las herramientas y fundamentación teórica pertinentes para determinar una posible solución que en este caso es la solución tecnológica tratada en este artículo. Posteriormente, con la información obtenida y la planteada en la fase de análisis se prosiguió con la fase de los requerimientos los cuales son listados a continuación:

4.1. Requerimientos

En esta fase se concluyó que el software debe:

- Permitirle al administrador conectarse al sistema utilizando un usuario y una contraseña.

- Validar el usuario y la contraseña para conceder los permisos requeridos para el administrador.
- Permitir gestionar administradores.
- Leer los puertos configurados y activos.
- Registrar cada 5 minutos las lecturas de puertos.
- Registrar las aplicaciones activas utilizadas por los puertos.
- Generar una vista de las lecturas de puertos para conocer las aplicaciones usadas permitiendo filtrar el listado generado por fechas y aplicaciones.

Después, se prosiguió con el diseño el cual fue basado en UML y se obtuvieron los siguientes resultados.

4.2. UML

4.2.1. Casos de uso

Se realizó el diagrama de casos de uso para describir que debe hacer el sistema en su interacción con los actores. Es decir, para describir el comportamiento observable externamente, sin entrar en la funcionalidad interna del sistema. (Pointeau, 30) El resultado es el siguiente:

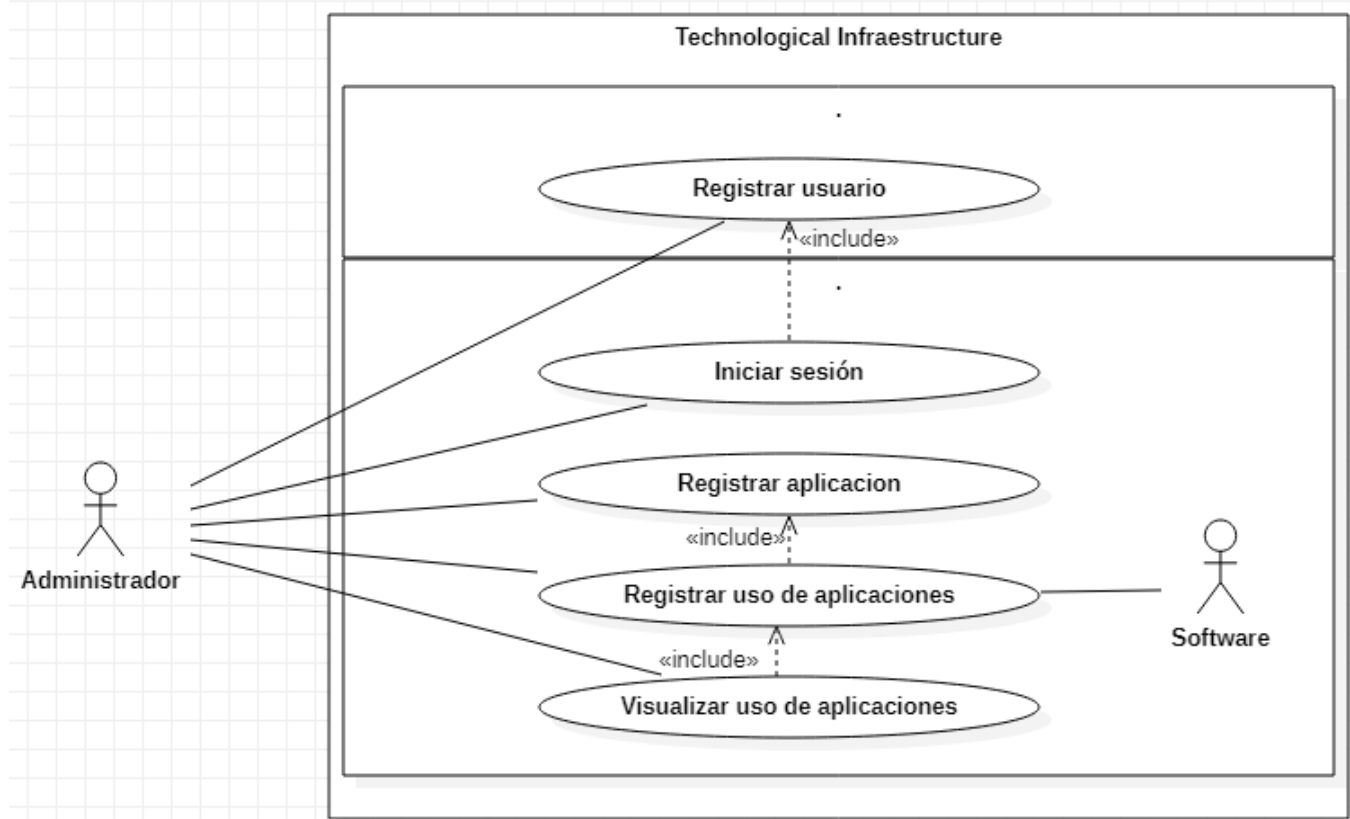


Figura 2. Diagrama de casos de uso

Se considera el caso de uso “Registrar usuario” como independiente debido a que es una acción poco frecuente con respecto a las otras.

4.2.2. Actividades

Se realizó el diagrama de actividades para describir los de procesos, se usó para modelar de forma gráfica los diferentes casos de uso, transacciones o procedimientos que haya la aplicación. En resumen, este diagrama sirvió para representar la forma en la que la aplicación hace una implementación. (Minguillón, 2010)

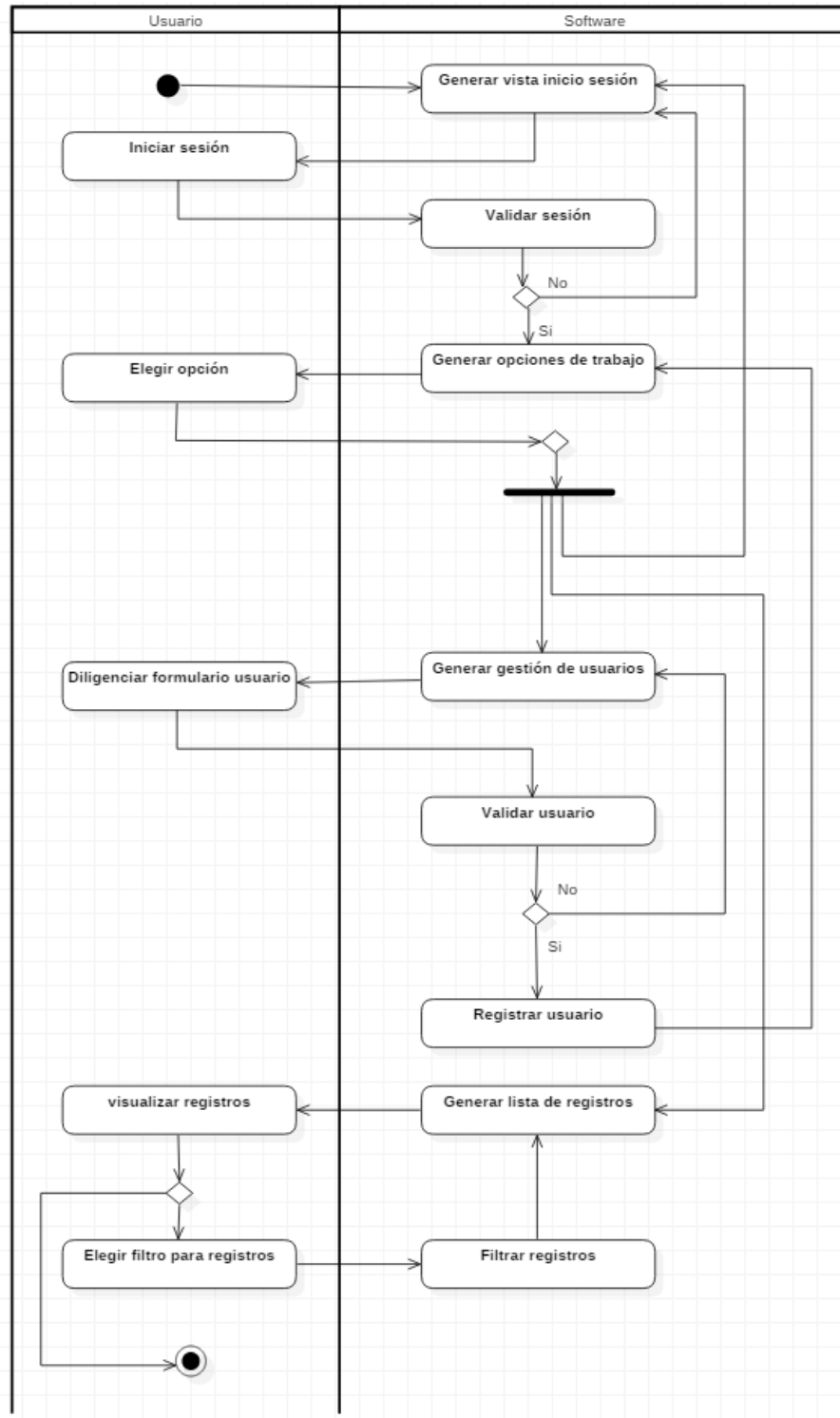


Figura 3. Diagrama de actividades

4.2.3. Clases

Se realizó el diagrama de clases como herramienta para comunicar el diseño de la aplicación que se creó en cuanto a objetos y las relaciones entre diferentes entidades. (Cifuentes, 2019) Para nuestro caso manejamos cuatro clases (Administrador, Aplicación, Registro y Equipo) como entidades y una clase (BD) como controlador la cual, nos brinda la conexión de nuestra aplicación con la base de datos.

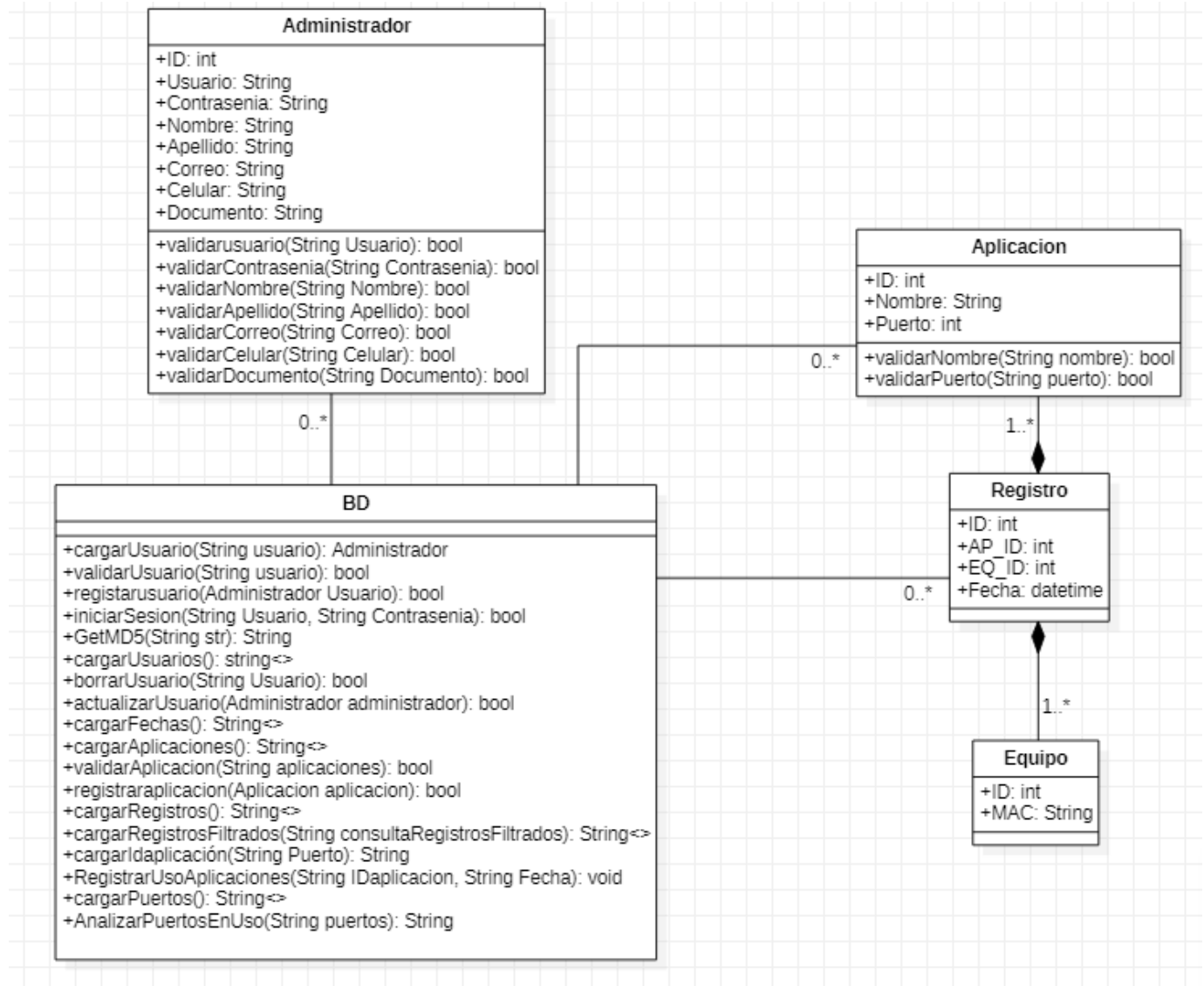


Figura 4. Diagrama de clases

4.2.4. Entidad Relación

Se realizó el modelo entidad como herramienta para representar de manera simplificada los componentes que participan y el modo en el que estos se relacionan entre sí a nivel de base de datos. (Hidalgo Pérez, 2017) Para nuestro caso manejamos cuatro entidades (Administradores, registros, aplicaciones y equipos) para el almacenamiento de los datos de estos.

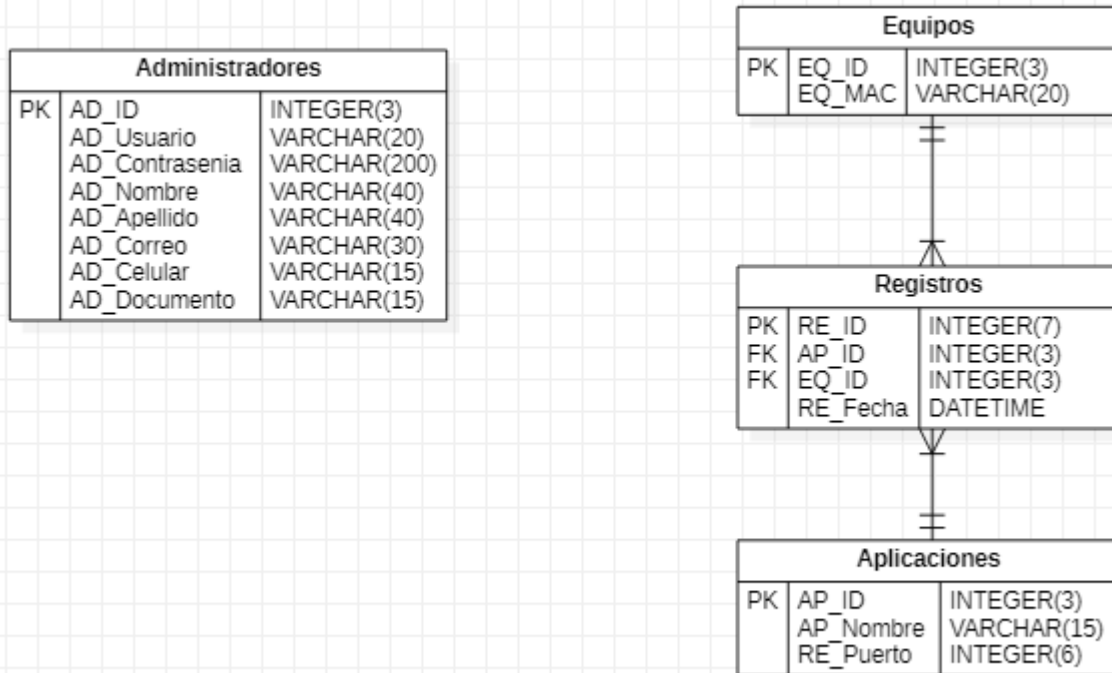


Figura 5. Diagrama entidad relación

4.3. Software

EL resultado final de este proyecto consiste, en una aplicación de escritorio desarrollada en lenguaje c# utilizando el framework .Net y Windows forms cuya función es hacer auditoría en los puertos de las aplicaciones registrados. Por medio de informes en los cuales se evidencia la fecha y hora capturas para cada aplicación en los registros realizados.

La aplicación realiza registros cada 5 minutos de forma automática y además permite hacer tomas manuales (Máximo 1 cada minuto). La aplicación cuenta con protección de acceso a ella por medio de la autenticación de usuario y contraseña. Además, permite que los usuarios administren otros usuarios (Crear, editar y eliminar).

Una vista más detallada a su funcionamiento se documenta a continuación:

4.3.1. Menú

Al iniciar sesión se dan las opciones al usuario de poder visualizar los registros, gestionar usuarios y cerrar sesión.

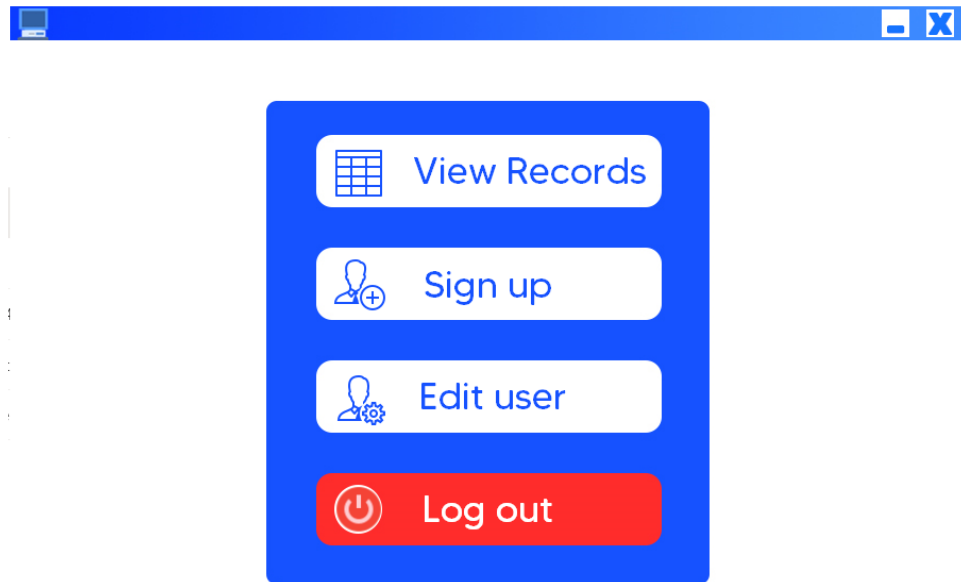


Figura 6. Menú de la aplicación.

4.3.2. Registrar usuarios

En la vista de registrar usuarios se proporcionan los campos para llenar la información de este, se manejan excepciones de campos vacíos, de usuarios ya existentes, de confirmación de contraseña. A nivel de base de datos, la llave primaria es un ID de registro debido a que las excepciones de valores repetidos se manejan por código.

A screenshot of the 'Sign Up' form in a mobile application. The form is a blue rectangle with rounded corners. At the top, there is a blue header bar with a back arrow icon on the left and the title 'Sign Up' in white. Below the header, there are eight input fields, each with a white icon on the left and a label: 'Name' (person icon), 'Last name' (person icon), 'Document' (ID card icon), 'Email' (envelope icon), 'Phone Number' (phone icon), 'Username' (person icon), 'Password' (key icon), and 'Confirm password' (key icon). At the bottom of the form is a blue button with the text 'Sign Up' in white.

Figura 7. Formulario de registro de usuarios

4.3.3. Gestión de usuarios

Cuando se ingresa a la edición de usuarios, se da la opción de elegir en lista los usuarios existentes y para el que se seleccione se ofrece la opción de actualizarlo o eliminarlo. Además, si se selecciona el usuario con el que se inició sesión no permitirá eliminarlo.

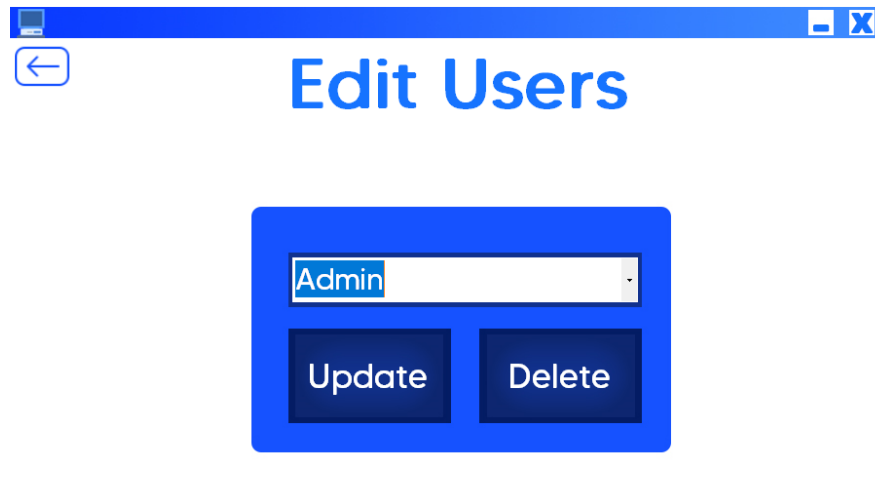


Figura 8. Opciones para la gestión de usuarios.

En caso de que se seleccione la edición de usuario, se despliega el formulario con la información del usuario seleccionado (a excepción de la contraseña) para que se efectúen los cambios requeridos y se manejan todas las excepciones presentadas en el registro de usuario.



Figura 9. Formulario para la actualización de usuarios.

4.3.4. Visualización de registros.

Cuando el usuario ingresa a visualizar los registros, se despliega la vista en la cual puede filtrar registros, visualizar la lista de registros (filtrados o no), agregar una aplicación para empezar a tener en cuenta en los nuevos registros y el botón que permite hacer un registro en tiempo real antes de que se haga automático cada 5 minutos.

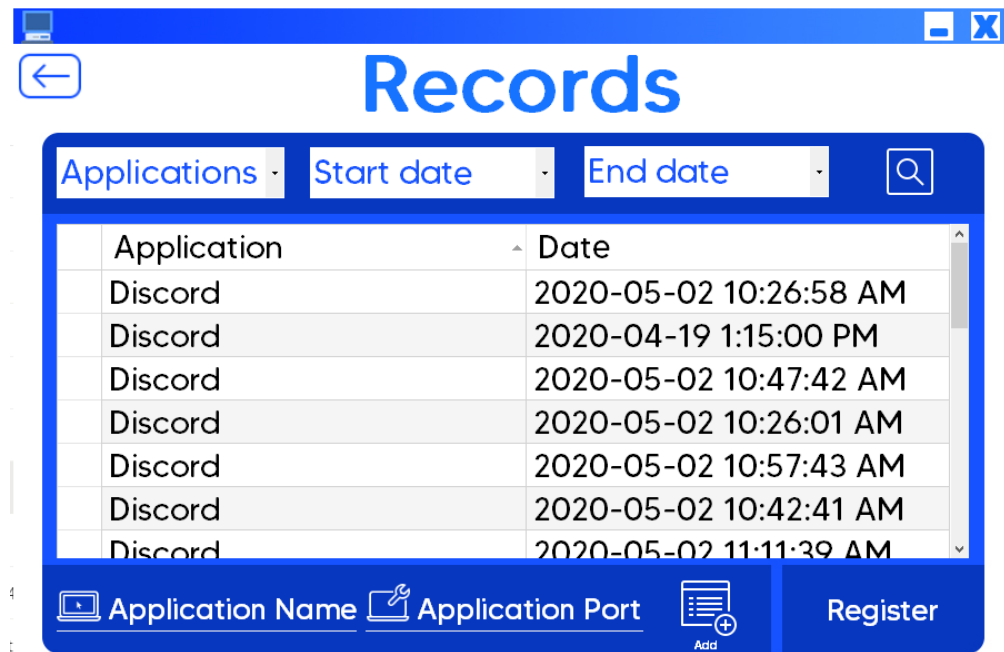


Figura 10. Visualización de registros

Para el filtro de los registros el usuario puede filtrar por dos aspectos fechas y/o aplicaciones como se detalla a continuación:

- Filtro de aplicaciones: En esta opción el usuario puede seleccionar si ver todas las aplicaciones o una aplicación en específico.
- Filtro de fechas: en esta opción el usuario puede seleccionar si ver todas las fechas, una sola fecha (posicionando la fecha inicial y la fecha final en la misma fecha) o un rango de fechas (posicionando los limites en el campo correspondiente).

Para agregar una nueva aplicación, el usuario debe colocar el nombre de aplicación y el puerto en los campos destinados y agregarlos con el botón “Add”. En esta opción se manejan las excepciones de valores repetidos por códigos y por ende no permite añadir una aplicación ya registrada.

El botón “Register” es el que permite realizar un registro por fuera de los registros automáticos, este botón se desactiva por un minuto al ser utilizado para evitar el uso desmesurado e ineficiente de los registros.

4.4. Pruebas

Para las pruebas de usuario se analizaron las siguientes situaciones con los datos de prueba mencionados en cada una:

4.4.1. Inicio de sesión

Se inicia sesión con una cuenta y contraseña incorrecta, el usuario rectifica datos y vuelve a intentar iniciar sesión.

#	Usuario	software	Datos	Revisar
1		Cargar formulario de inicio de sesión		
2	Ingresa usuario y contraseña		Admi 12345	
3		Arrojar mensaje de cuenta incorrecta y limpia el campo contraseña		Rectifica datos
4	Ingresa usuario y contraseña		Admin 123456	
5		Concede el acceso y ejecuta menú		
#	Errores - Descripción		Corregido	fecha de corrección

Tabla 1. Prueba inicio de sesión

4.4.2. Registrar Usuario

Registra un usuario ya existente y correo electrónico incorrecto.

#	Usuario	software	Datos	Revisar
1		Cargar formulario de registro de usuarios		
2	Ingresa: Nombre Apellido Documento Correo Celular Usuario Contraseña Confirmación de contraseña		Karen Luna 1098754986 Karen.com 3215469874 Admin 123456 123456	
3		Arrojar mensaje correo incorrecto y devuelve al formulario limpiando las contraseñas.		Rectifica datos

4	Ingresa: Nombre Apellido Documento Correo Celular Usuario Contraseña Confirmación de contraseña		Karen Luna 1098754986 Karen@hotmail.com 3215469874 Admin 123456 123456	
5		Crea el usuario y devuelve al menú		
#	Errores - Descripción		Corregido	fecha de corrección
1	El software permite crear usuarios con el mismo nombre de usuario		Si	6/04/2020

Tabla 2. Prueba registro de usuario

4.4.3. Registrar aplicación

El usuario agrega una aplicación ya registrada

#	Usuario	software	Datos	Revisar
1		Cargar formulario de registro de aplicación		
2	Ingresa: Aplicación Puerto		Discord 64036	
3		Arrojar mensaje aplicación ya registrada y limpia campos		Rectifica datos
4	Ingresa: Aplicación Puerto		Chrome 5353	
5		Arrojar mensaje registro exitoso		
#	Errores - Descripción		Corregido	fecha de corrección

Tabla 3. Prueba registro de aplicación

4.4.4. Eliminar usuario

El usuario elimina la cuenta con la que se inicia sesión

#	Usuario	software	Datos	Revisar
1		Cargar combo con usuarios		
2	seleccionar usuario: Admin		Admin	
3		Elimina Usuario y devuelve al menu		
4				
5				
#	Errores - Descripción		Corregido	fecha de corrección
1	El software permite eliminar la cuenta con la que se inició sesión		Si	6/04/2020

Tabla 4. Prueba eliminación de usuario

Adicionalmente, se realizaron pruebas de:

- Filtros en los registros lo cuales inicialmente no permitían analizar rangos de fecha si en la fecha inicial se colocaba una fecha mayor a la final, esto se corrigió el 6/04/2020.
- El botón de registrar inicialmente no se bloqueaba y permitía realizar registros desmesuradamente, esto se corrigió el 6/04/2020.
- Las excepciones de permitir letras en los campos de documento y teléfono, y permitir números en los nombres y apellido se corrigieron el 7/04/2020 manejando la excepción desde el textBox correspondiente.

5. CONCLUSIONES

El análisis del problema permitió que el proceso fuera acertado debido a que se fijó correctamente la solución al problema y requerimientos adecuados. Además, la fijación del alcance del proyecto, aunque se vio afectada por aspectos ajenos fue de gran ayuda para facilitar el desarrollo de la aplicación.

El uso de UML fue importante en el proceso porque con esto se puede tener claridad del comportamiento de la aplicación y de esta forma desde el diseño ir planeando la implementación.

La metodología brindó la flexibilidad requerida para que la implementación del software fuera más ágil debido a que durante esta etapa se pudieron hacer las modificaciones pertinentes para acoplar la implementación a los requerimientos y el diseño.

Las pruebas de usuario permitieron un mejoramiento significativo en el resultado final. Sin embargo, el proceso de realimentación por etapas permitió mitigar errores durante el desarrollo y dejar muy pocos en el resultado.

Referencias

- Carvajal, R., Gómez, U., & Albarracín, E. (2020). *Definición del Proyecto Integrador Facultad de Ingeniería de Sistemas e Informática Primer semestre de 2020*. Bucaramanga: Universidad Pontificia Bolivariana.
- Cifuentes, J. (2019 de Abril de 2019). *Diagrama de Clases*. Obtenido de [manuel.cillero.es: https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-clases/](https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-clases/)
- Domínguez, P. (6 de Febrero de 2020). *En qué consiste el modelo en cascada*. Obtenido de <https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada>
- Hidalgo Pérez, L. (3 de Julio de 2017). *Modelo entidad relación: descripción y aplicaciones*. Obtenido de ICEMD: Modelo entidad relación: descripción y aplicaciones
- Krall, C. (26 de Octubre de 2018). *¿Qué es y para qué sirve UML? Versiones de UML (Lenguaje Unificado de Modelado). Tipos de diagramas UML*. Obtenido de [aprenderaprogramar.com: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:que-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46&Itemid=163](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:que-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46&Itemid=163)
- Minguillón, J. (2010). *Introducción al Lenguaje de Modelado Unificado*. FUOC.
- Pointeau, A. (2018 de Marzo de 30). *Diagramas de Casos de Uso*. Recuperado el 1 de Mayo de 2020, de <https://openclassrooms.com/en/courses/4990961-planea-tu-proyecto-con-uml/4996511-diagramas-de-casos-de-uso>
- Puntero Méndez, Á. (10 de Agosto de 2019). *¿Qué puertos usa cada aplicación?* Obtenido de PC actual: https://www.pcactual.com/noticias/trucos/puertos-cada-aplicacion-2_2845
- Raffino, M. (14 de Febrero de 2020). *Base de datos*. Obtenido de <https://concepto.de/base-de-datos/>
- RedHat. (2020). *¿Qué es el open source?* Obtenido de <https://www.redhat.com/es/topics/open-source/what-is-open-source>
- Saavedra, A. (2018 de Febrero de 2018). *Clave i*. Recuperado el 3 de Febrero de 2020, de <https://www.clavei.es/blog/que-es-la-infraestructura-it/>
- Schiaffarino, A. (12 de Marzo de 2019). *Modelo cliente servidor*. Obtenido de [infranetworking: https://blog.infranetworking.com/modelo-cliente-servidor/](https://blog.infranetworking.com/modelo-cliente-servidor/)