



UNIVERSIDAD DE ARTES, CIENCIAS Y COMUNICACIÓN
Facultad de Administración
Carrera de Ingeniería Informática Multimedia

Título del trabajo

S.I.A.O.
(Sistema Integrado Alertas Oportunas)

Profesor Guía:

Felipe Alejandro Montenegro González

Estudiantes:

Rolando Andrés Bergmann Pino

Marco Antonio Donoso Díaz

Juan José Ortiz Palma

Santiago de Chile, abril de 2022

Índice

Índice	2
Introducción	4
Planteamiento del Problema	5
Referentes Teóricos	6
Factibilidad del proyecto	7
Factibilidad técnica	7
Factibilidad económica	7
Factibilidad legal	8
Factibilidad de recursos	8
Factibilidad de mercado	8
Factibilidad operacional	8
Factibilidad de tiempo	8
Metodología Utilizada	9
Diseño del Software	10
Conceptos relevantes y/o Supuestos	10
Diccionario de Entidades	11
Diagrama Entidad Relación de Base de Datos	12
Modelo Entidad Relación de Base de Datos	13
Diccionario de Datos	14
Pruebas	24
Pruebas de caja blanca y pruebas de caja negra	24
Desarrollo del Caso de Uso	26
Pruebas de Caja Negra	28

Pruebas de Caja blanca	31
Observaciones sobre los resultados obtenidos	32
Conclusión	33
Recomendaciones	34
Referencias	35

Introducción

La necesidad de establecer sistemas de comunicación de alertas de vehículos motorizados que transportan materias primas se ha hecho altamente necesaria dada la contingencia nacional en materia delictual, en atención al alto flujo de vehículos transitando por las rutas y por la necesidad del cumplimiento de las actividades contractuales de dichas empresas transportadoras.

Por lo antes expuesto, es que se ha creado el sistema de alarmas SIAO, que tiene por objetivo alertar a la empresa contratante respecto a atentados, robos e incendios que puedan sufrir sus vehículos de transportes.

En atención al servicio que este sistema puede prestar, es que la empresa SOFTE SPA que presta servicios a la empresa Arauco, que genera productos para construcción y mueblería, con presencia en 5 continentes, ha contratado e implementado el sistema de alertas SIAO.

El presente trabajo tiene por objeto representar el funcionamiento y ejecución del sistema SIAO, desde el momento en que el usuario genera una alarma hasta que ésta es resuelta.

Considerando las etapas correspondientes al desarrollo de dicho sistema hasta su integración, implementación y producción.

Planteamiento del Problema

Este es un sistema que tiene por objeto dar solución a un requerimiento específico; en este caso en particular poder reaccionar de la manera más rápida posible a eventos como: atentados, robos e incendios, ya sea para que personal de la empresa concurra a verificar los hechos alertados y a su vez contactar a las autoridades correspondientes si es que fuese necesario.

El contexto reviste ciertas particularidades debido a que la empresa Arauco es una de las empresas de transportes de madera que ha sido víctima en innumerables ocasiones de atentados en la VIII y IX Región del País, lo que sumado a los robos e incendios forestales hace que sea imperativo que este sistema funcione cumpliendo todas las exigencias de la empresa afectada.

Algunos entes externos interesados son Carabineros, Bomberos y Conaf, ya que gracias a este sistema permite reportar de una manera más rápida los eventos descritos anteriormente, siguiendo los protocolos preestablecidos.

Es entonces que Arauco se contacta con SOFTE SPA, para que esta empresa implemente mejoras a su sistema actual de alertas. Ya que la empresa Arauco esta disconforme con el actual sistema de alerta que poseen, toda vez que su sistema actual carece de la fiabilidad, debido a múltiples errores, fallas y estabilidad presentadas por este.

Es por este motivo que el sistema que hemos creado se ha implementado, el cual presenta usabilidad, tiempos oportunos de respuesta, portabilidad, debido a que se puede utilizar tanto en computadores con acceso a internet, como en teléfonos y tablets con sistema operativo Android con acceso a internet.

Dada las condiciones geográficas, políticas, económicas y demográficas del lugar donde se encuentra emplazada la empresa Arauco, se hizo imperativo la implementación de un sistema que cumpliera con todas las características antes mencionadas.

Referentes Teóricos

En el mercado actual existen variados precedentes de sistemas y empresas que prestan servicios similares, los cuales permiten tener sistemas de Alertas respectivas de seguridad, como lo es el sistema SOSAFE o las empresas de seguridad ADT o VERISURE.

Dicho sistema y empresas nos permiten poder alertar distintos tipos de eventos, en el caso del sistema SOSAFE, permite alertar sobre eventos delictuales que están ocurriendo en un sector urbano específico.

En el caso de las empresas de seguridad, estas entregan servicios de seguridad específicos a domicilios particulares y empresas, alertando a los usuarios y a las autoridades respectivas, sobre eventos delictuales.

El sistema SIAO el cual ya está implementado en la empresa Arauco, tiene el mismo objetivo, el cual consiste en dar alertas tempranas sobre posibles eventos delictuales y acontecimientos naturales, que puedan afectar el correcto funcionamiento de los procesos de logística de transporte de materias primas.

Con la diferencia que este sistema se encuentra en línea con los vehículos y los usuarios de estos, reportan constantemente a la central respectiva sobre posibles alertas de eventos.

El sistema SIAO, se ajusta a lo establecido en la ley n°19.628 sobre la protección de la vida privada.

Factibilidad del proyecto

Respecto a la factibilidad del sistema SIAO, éste ya había sido aprobado antes de ser ejecutado, no obstante, a esto, con el fin de poder estudiar su rentabilidad respecto a otros clientes, se realizaron las siguientes factibilidades.

Factibilidad técnica

Para la implementación del sistema se utilizó el hardware que posee el cliente:

- Central de monitoreo completamente equipada
- Tablets, los cuales serán usados por los camioneros en la ruta.
- Teléfonos celulares
- Botones de pánico, los cuales serán usados por rondines dentro de la empresa.

Junto con lo descrito anteriormente se requiere de internet móvil para los tablets y teléfonos celulares y de internet fijo para la central.

Tanto el alojamiento de la aplicación y de la base de datos es en servidores virtuales (VPS). Se planea en un futuro implementar todo en Microsoft Azure.

Factibilidad económica

Al no tener acceso a los costos que le generan a Arauco actualmente los eventos descritos en la descripción del proyecto, pero considerando el valor de la maquinaria que poseen es bastante elevado.

El costo mensual es de 30 UF, cualquier módulo que la empresa requiera posterior a la entrega, es un cargo extra, y el monto depende de la complejidad del desarrollo e implementación lo cual ya fue aprobado por Arauco.

Factibilidad legal

En la actualidad no contamos con los requisitos legales ya que de la empresa Arauco para la implementación del sistema, ya que la contratación del servicio es a través de una empresa externa que revisa estos.

Factibilidad de recursos

En este sentido no fue necesario ahondar, toda vez que el personal ya existe y los recursos tecnológicos también.

Factibilidad de mercado

De seguir siendo exitosa la implementación en la empresa Arauco existen grandes posibilidades de expandir este servicio a otras empresas, organizaciones sociales y particulares.

Factibilidad operacional

Por lo descrito en otros puntos se hace evidente que es imperativo para la empresa poder tener la capacidad de reportar incidentes en tiempo real, es este el fin de la implementación y por qué la empresa solicitó ayuda de SOFTE SPA.

Factibilidad de tiempo

En cuanto al tiempo de duración del proyecto esto no presenta una gran complicación, ya que gracias al uso de la metodología SCRUM se presentará un MVP funcional, que de ser necesario puede ser utilizado durante el desarrollo.

Metodología Utilizada

Para el desarrollo del software se utilizó la metodología (framework) ágil SCRUM. En SCRUM el trabajo se divide en Sprints, periodos de tiempo cortos generalmente de 2 semanas, donde se planea una cantidad determinada de trabajo a completar.

Cada Sprint consta de 5 partes:

1. Planear el Sprint: Se define el objetivo del Sprint.
2. Scrum diario: Reunión diaria de no más de 15 minutos donde el equipo se pone al día con lo que va haciendo el resto y se pide y ofrece ayuda si es necesario.
3. Desarrollo durante el Sprint: Es cuando se realiza el desarrollo por parte del equipo, siempre el resultado del Sprint es el de un software usable y un avance sobre el resultado del Sprint pasado.
4. Revisión del Sprint: Se revisa los resultados obtenidos durante el Sprint.
5. Retrospectiva del Sprint: Se analiza lo bueno y lo malo que ocurrió durante el Sprint, oportunidades de mejora, etc.

Diseño del Software

El contexto del sistema SIAO se enmarca en lo que es la seguridad empresarial, más específicamente en la seguridad de la operación de la empresa Arauco.

No obstante, a esto, el sistema está diseñado para ser empleado tanto en seguridad empresarial, como particular, debido a que el modelo de negocio del sistema permite acceder a diversos clientes y sus diversas necesidades, ya sean previos o fondos particulares, como comunidades sociales (juntas de vecinos) e inclusive poder trabajar con municipios.

Si bien se considera que el sistema puede expandir su capacidad de clientes y adaptarse a sus múltiples necesidades, su función principal es la seguridad de empresas ya sean privadas o estatales.

Respecto al desarrollo de la base de datos del sistema SIAO, se indica lo siguiente:

Conceptos relevantes y/o Supuestos

- Portador: Quien genera las alarmas y es capaz de enviar señales de alerta desde su dispositivo a la central.
- Señal de alerta: Enviada por el portador a la central utilizando un dispositivo. Se categorizan como eventos reales o de prueba
- Central: Donde se reciben y procesan las señales de alarma.
- Operario de la Central: Quién debe hacer el procesamiento, confirmación y actuar cuando sea necesario en base a las alarmas recibidas
- Dispositivos: Estos pueden ser tablets, celulares o botones de pánico, los cuales utilizando tecnología celular 3G o 4G envían las señales a la central.
- App: Una pequeña aplicación que permite al usuario enviar alertas a la central, tanto de pruebas o reales.

- Rondín: Portador de dispositivo que debe hacer rondas y generar alarmas cuando sea necesario.
- Ronda: Camino que debe recorrer el Rondín.
- Vehículo: Medio de transporte registrado por la empresa. Su conductor o acompañante son quienes deben generar las alarmas de ser necesario.
- Chofer: Es quien maneja uno de los vehículos de la empresa, portador de dispositivo que puede generar alarmas.

Diccionario de Entidades

Empresa: Es la entidad que describe la empresa que será ingresada al sistema.

Usuario: es la entidad que representa a los usuarios del sistema.

Auditoría: Es la entidad necesaria para auditar los botones (dispositivos) ingresados al sistema.

contactoEmergencia: esta entidad tiene los contactos de emergencia del portador.

Vehículo: esta entidad describe los vehículos que serán ingresados al sistema.

Portador: esta entidad describe al usuario portador del sistema

Dispositivo: esta entidad describe los dispositivos del sistema.

Evento: esta entidad describe un evento, es donde se almacenarán las señales de alarma.

DispositivoCoordenada: esta entidad describe las coordenadas del dispositivo.

Coordenada:

Ronda: Esta entidad describe la ronda que será ingresada al sistema.

Diagrama Entidad Relación de Base de Datos

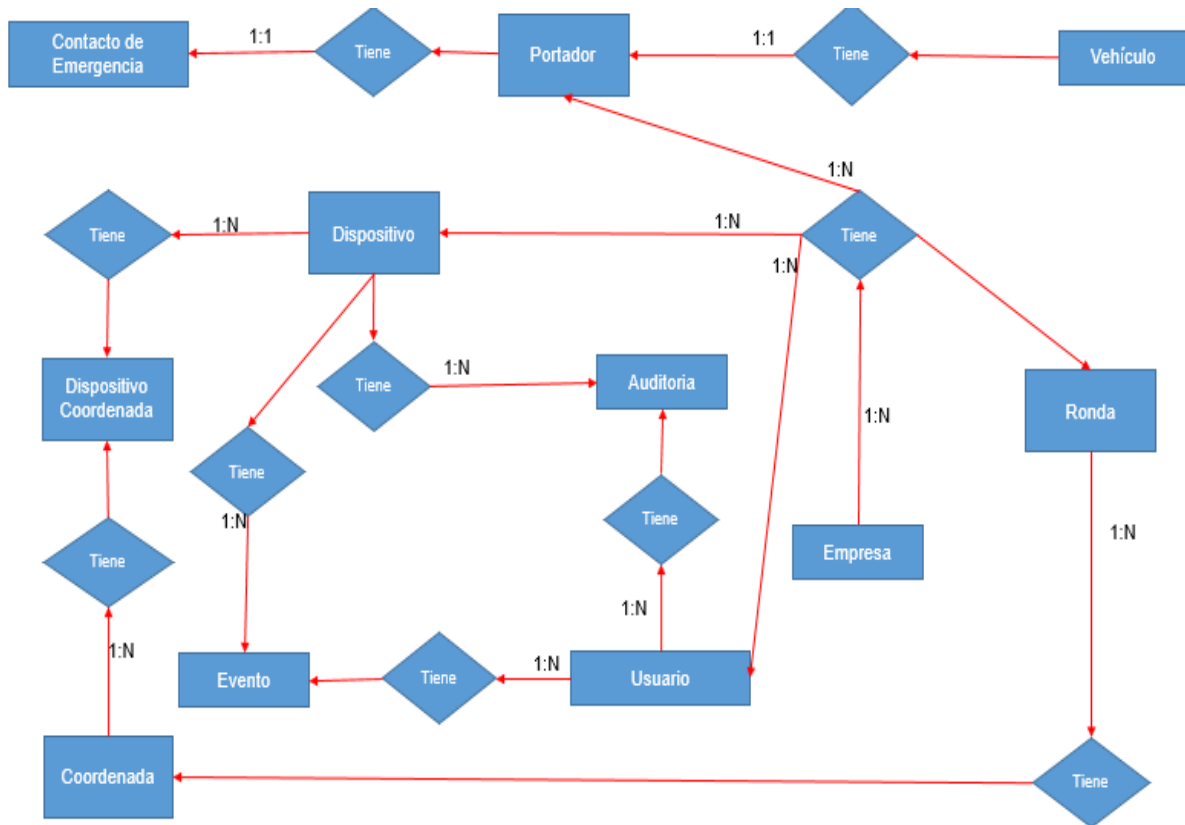


Imagen: Diagrama Entidad Relación

Modelo Entidad Relación de Base de Datos

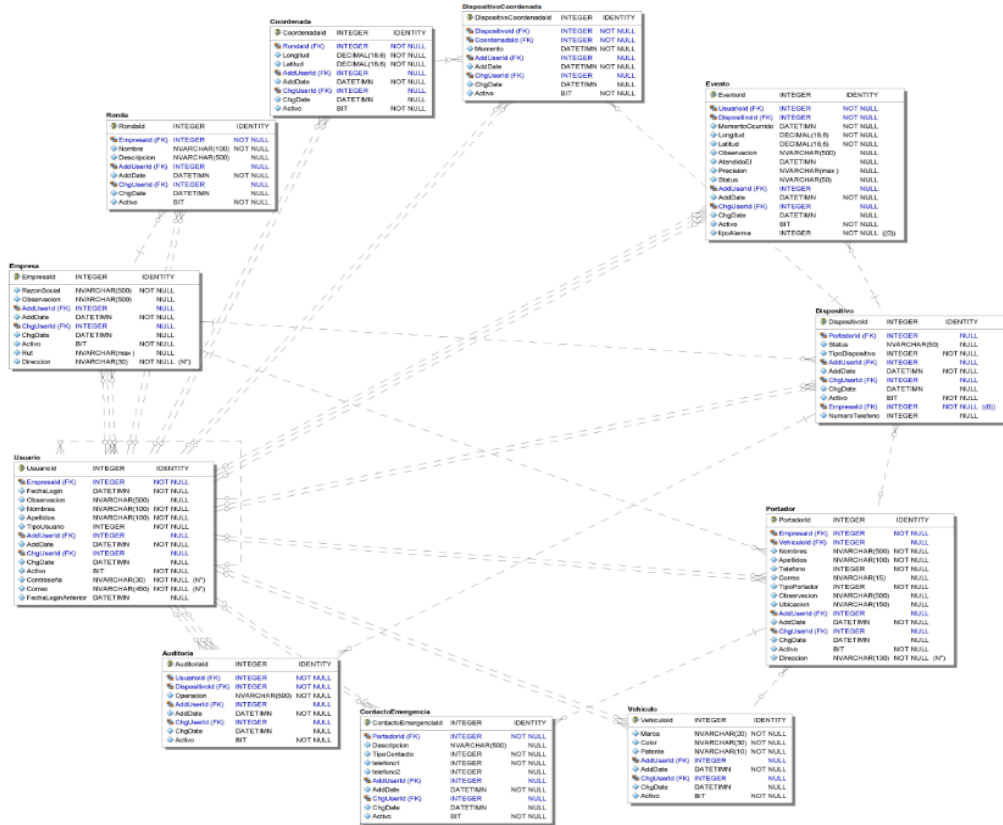


Imagen Modelo Entidad Relación

Diccionario de Datos

Entidad Empresa

1.- empresarialID tipo de dato integer, not null, clave primaria, identificador numérico único de cada empresa que contará con la aplicación.

2.- razonSocial tipo de dato nvarchar(500), not null, Identificador, el cual como hace su nombre hace referencia a la razón social de la empresa que contrata la aplicación.

3.- Observacion tipo de dato nvarchar(500), null, campo para agregar alguna observación de la empresa.

3.- addUserID tipo de dato integer, null, identificador de usuario del sistema, quien es el que genera algún cambio dentro de la tabla de identidad.

4.- addDate tipo de dato datetime, not null, fecha ingreso del registro.

5.- chgUserID tipo de dato integer, null, identificador usuario que realiza modificación al registro

6.- chgDate tipo de dato datetime, null, fecha modificación del registro.

7.- activo: tipo de dato bit, not null, estado actual de la empresa con respecto al contrato vigente con la aplicación.

8.- rut tipo de dato nvarchar(max), null, rut de la empresa que contrata la aplicación.

9.- dirección tipo de dato nvarchar(30),not null, dirección de la empresa que contrata la aplicación.

Entidad Usuario

1.- UsuarioID tipo de dato integer, not null, clave primaria, identificador único de cada usuario que utiliza la aplicación.

2.- empresald tipo de dato integer, not null, clave foránea correspondiente a la relación de la tabla usuario con empresa.

3.- fechaLogin tipo de dato datetime, not null, identificador de la fecha de inicio de sesión por parte del usuario en la aplicación.

4.- observación tipo de dato nvarchar(500), null, campo para agregar alguna observación del usuario.

5.- nombres tipo de dato nvarchar(100), not null, nombres del usuario que utiliza la aplicación.

6.- apellidos tipo de dato nvarchar(100), not null, apellidos del usuario que utiliza la aplicación.

7.- tipoUsuario tipo de dato integer, not null, categoría del usuario, la cual determina el rol de este en el uso de la aplicación.

8.- addUser tipo de dato integer, null, identificador de usuario del sistema, quien es el que genera algún cambio dentro de la tabla de identidad.

9.- chgUserId tipo de dato integer, not null, identificador del usuario que modifica el registro

10.- addDate tipo de dato datetime, not null, fecha ingreso del registro.

11.- chgDate tipo de dato datetime, null, fecha modificación del registro.

12.- activo tipo de dato bit, not null, estado del usuario, cuando este se encuentra utilizando la aplicación.

13.- contraseña tipo de dato nvarchar(30), not null, contraseña con la cual se corrobora o verifica al usuario antes de entrar a la aplicación.

14.- correo tipo de dato nvarchar(450), not null, correo de contacto del usuario, el cual a su vez nos permite poder enviar mensajes de estado de cuenta y mensajes relacionados con el estado de la aplicación.

15.- fechaLoginAnterior tipo de dato datetime, null, indica cuál fue la última vez que ingresó el usuario a la aplicación.

Entidad Auditoria

1.- auditorialID tipo de dato integer, clave primaria, identificador numérico único de cada auditoría que se realiza a la aplicación, con respecto al control de modificaciones que se hacen en las entidades correspondientes.

2.- usuarioid tipo de dato integer, not null, clave foránea correspondiente a la relación de la tabla auditoria con usuario.

3.- dispositivoid tipo de dato integer, not null, clave foránea correspondiente a la relación de la tabla auditoria con dispositivo.

4.- operación tipo de dato nvarchar (500), not null, indica que tipo de acción se realizó dentro de la tabla.

5.- addDate tipo de dato datetime, not null, fecha ingreso del registro.

6.- chgUserId tipo de dato integer, null, identificador de usuario que modifica el registro.

7.-chgDate tipo de dato datetimn, null, fecha modificación registro.

8.- activo tipo de dato bit, not null, estado de la acción realizada dentro de la tabla.

Entidad contactoEmergencia

1.- contactoEmergenciald tipo de dato integer, clave primaria, identificador único década contacto de emergencia dentro de la tabla.

2.- portadorId tipo de dato integer, not null, clave foránea correspondiente a la relación de la tabla contactoEmergencia con portador.

3.- descripción tipo de dato nvarchar(500), null, descripción del contacto de emergencia, donde se especifica información descriptiva de este.

4.- tipoContacto tipo de dato integer, not null, especifica el tipo del contacto en la tabla.

5.- telefono1 tipo de dato integer, not null, primer número telefónico del contacto de emergencia.

6.- telefono2 tipo de dato integer, null, segundo número telefónico del contacto de emergencia.

7.-addUserId tipo de dato integer, null, identificador de usuario del sistema, quien es el que genera algún cambio dentro de la tabla de identidad.

8.- addDate tipo de dato datetime, not null, fecha ingreso del registro.

9.- chgUserId tipo de dato integer, null, identificador del usuario que modifica el registro.

10.-activo tipo de dato bit, not null, estado del contacto de emergencia dentro de la tabla.

Entidad Vehículo

1.- vehiculoid tipo de dato integer, clave primaria, identificador único de cada vehículo dentro de la tabla.

2.- marca tipo de dato nvarchar(20), not null, marca del vehículo.

3.- color tipo de dato nvarchar(30), not null, color del vehículo.

4.- patente tipo de dato nvarchar(10) not null, registro único nacional del vehículo.

5.- addUserId tipo de dato integer, null, identificador de usuario del sistema, quien es el que genera algún cambio dentro de la tabla de identidad.

6.- addDate tipo de dato datetime, null, fecha ingreso registro.

7.- chgUserId tipo de dato integer, null, identificador usuario que modifica el registro.

8.- chgDate tipo de dato datetime, null, fecha modificación del registro.

9.- activo tipo de dato bit, not null, estado del vehículo dentro de la tabla.

Entidad Portador

1.- portadorId tipo de dato integer, clave primaria, identificador único del portador dentro de la tabla.

2.- empresald tipo de dato integer, not null, clave foránea que relaciona la tabla portador con la tabla empresa.

3.- vehiculold tipo de dato integer, null, clave foránea que relaciona la tabla portador con la tabla vehículo.

4.- nombres tipo de dato nvarchar(500), not null, nombres del portador asignado.

5.- apellidos tipo de dato nvarchar(100), not null, apellidos del portador asignado.

6.- teléfono tipo de dato integer, not null, teléfono del portador asignado.

7.- correo tipo de dato nvarchar(15), null, correo de contacto del portador asignado.

8.- tipoPortador tipo de dato integer, not null, categoría de portador asignado.

9.- observación tipo de dato nvarchar(500), null, observación realizada por portador asignado.

10.- ubicación tipo de dato nvarchar(150), null, indica la ubicación actual del portador.

11.- addUserId tipo de dato integer, null, identificador del usuario del sistema, quien es el que genera algún tipo de cambio en la tabla portador.

12.- addDate tipo de dato datetime, not null, fecha ingreso registro.

13.- chgUserId tipo de dato integer, null, identificador usuario que modifica registro.

14.- chgdate tipo de dato datetime, null, fecha modificación registro.

15.- activo tipo de dato bit, not null, estado del portador dentro de la tabla.

16.- dirección tipo de dato nvarchar(130) not null, indica la dirección del portador en la tabla.

Entidad Dispositivo

1.- dispositivoid tipo de dato integer, null, clave primaria, identificador único del dispositivo dentro de la tabla.

2.- portadorId tipo de dato integer, null, clave foránea que relaciona la tabla dispositivo con portador.

3.- tipoDispositivo tipo de dato integer, not null, indica la categoría del dispositivo dentro de la tabla.

4.- addUserId tipo de dato integer, null, identificador del usuario del sistema, quien es que genera algún tipo de cambio en la tabla dispositivo.

5.- addDate tipo de dato datetime, not null, fecha ingreso del registro.

6.- chgUserId tipo de dato integer, null, identificador usuario que modifica el registro.

7.- chgdate tipo de dato datetime, null, fecha modificación registro.

8.- activo tipo de dato bit, not null, estado del dispositivo dentro de la tabla.

9.- empresald tipo de dato integer, not null, clave foránea que relaciona la tabla dispositivo con empresa.

10.- numerotelefono tipo de dato integer, null, número de teléfono del dispositivo de la tabla.

Entidad Evento

1.- eventold tipo de dato integer, clave primaria, identificador único del evento dentro de la tabla.

2.- usuariold tipo de dato integer not null, clave foránea que relaciona la tabla evento con usuario.

3.- dispositivold tipo de dato integer not null, clave foránea que relaciona la tabla evento con dispositivo.

4.- momentoOcurrido tipo de dato datetime, not null, indica en que momento se crea el evento.

5.- longitud tipo de dato decimal (18,6), not null, indica la longitud donde ocurre el evento.

6.- latitud tipo de dato decimal (18,6), not null, indica la latitud donde ocurre el evento.

7.- observación tipo de dato nvarchar(500), null, ingreso de observación o detalle del evento ocurrido.

8.- atendidoEl tipo de dato datetime, null, indica la fecha de atención del evento.

9.- precision tipo de dato nvarchar, null, precisión de la coordenada.

10.- status tipo de dato nvarchar(50), null, estado del registro.

11.- addUserId tipo de dato integer, null, identificador del usuario del sistema, quien es que genera algún tipo de cambio en la tabla evento.

12.- addDate tipo de dato datetime, not null, fecha ingreso registro.

13.- chgUserId tipo de dato integer, null, identificador usuario que modifica el registro.

14.- chgdate tipo de dato datetime, null, fecha modificación registro.

15.- activo tipo de dato bit, not null, estado del dispositivo dentro de la tabla.

16.- tipoalarma tipo de dato integer, not null, indica el tipo de alarma dentro del evento.

Entidad DispositivoCoordenada

1.- dispositivoCoordenadald tipo de dato integer, clave primaria, identificador único del dispositivo coordenada dentro de la tabla.

2.- dispositivold tipo de dato integer, not null, clave foránea que relaciona la tabla dispositivoCoordenada con dispositivo.

3.- coordenadald tipo de dato integer, not null, clave foránea que relaciona la tabla dispositivoCoordenada a coordenada.

4.- momento tipo de dato datetime, not null, indica el momento en que se ingresa la coordenada.

5.- addUserId tipo de dato integer, null, identificador del usuario del sistema, quien es que genera algún tipo de cambio en la tabla dispositivoCoordenada.

6.- addDate tipo de dato datetime, not null, fecha ingreso registro.

7.- chgUserId tipo de dato integer, null, identificador usuario que modifica el registro.

8.- chgdate tipo de dato datetime, null, fecha modificación registro.

9.- activo tipo de dato bit, not null, estado del dispositivo dentro de la tabla.

Entidad Coordinada

1.- coordenadald tipo de dato integer, clave primaria, identificador único de la coordenada dentro de la tabla.

2.- rondald tipo de dato integer, not null, clave foránea que relaciona la tabla coordenada con ronda.

3.- longitud tipo de dato decimal (18,6), not null, indica la longitud donde ocurre el evento.

4.- latitud tipo de dato decimal (18,6), not null, indica la latitud donde ocurre el evento.

5.- addUserId tipo de dato integer, null, identificador del usuario del sistema, quien es que genera algún tipo de cambio en la tabla coordenada.

6.- addDate tipo de dato datetime, not null, fecha ingreso registro.

7.- chgUserId tipo de dato integer, null, identificador de usuario que modifica el registro.

8.- chgdate tipo de dato datetime, null, fecha modificación registro.

9.- activo tipo de dato bit, not null, estado del dispositivo dentro de la tabla.

Entidad Ronda

1.- rondald tipo de dato integer, clave primaria, identificador único de la ronda dentro de la tabla.

2.- empresald tipo de dato integer, clave foránea que relaciona la tabla ronda con empresa.

3.- nombre tipo de dato nvarchar (100), not null, nombre de la ronda a efectuarse.

4.- descripción tipo de dato nvarchar (500), null, descripción de la ronda a efectuarse.

5.- addUserId tipo de dato integer, null, identificador del usuario del sistema, quien es que genera algún tipo de cambio en la tabla ronda.

6.- addDate tipo de dato datetime, not null, fecha de ingreso registro.

7.- chgUserId tipo de dato integer, null, identificador de usuario que modifica el registro.

8.- chgdate tipo de dato datetime, null, fecha modificación del registro.

9.- activo tipo de dato bit, not null, estado del dispositivo dentro de la tabla.

Pruebas

Pruebas de caja blanca y pruebas de caja negra

Respecto a las pruebas de caja blanca y negra en el sistema, estas fueron realizadas en base a la mantención básica de usuarios, también conocido como CRUD.

En primer lugar, dentro del sistema podemos observar la lista de usuarios que tenemos en nuestra aplicación, como se muestra en la siguiente imagen.

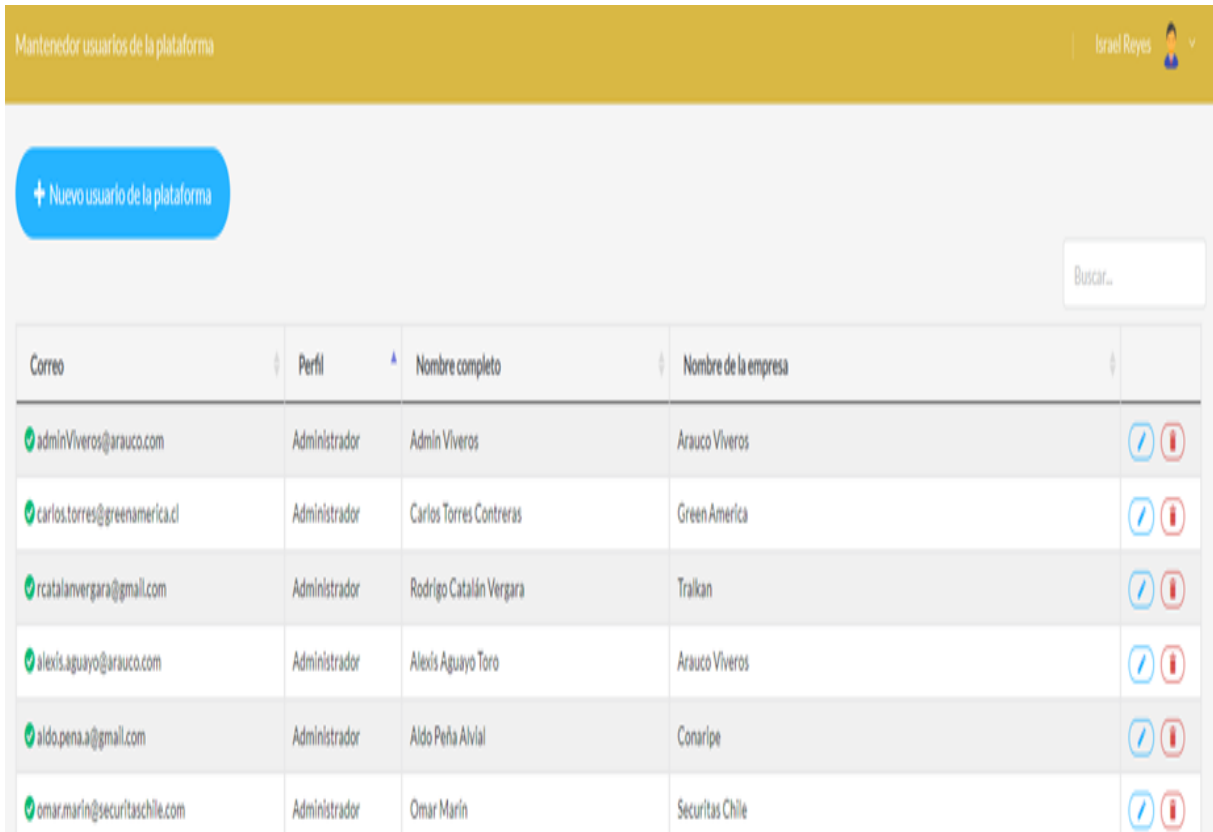




Imagen: Lista de Usuarios

Para el caso de uso respectivo, se muestra la imagen donde se nos presenta la ventana respectiva a la creación de un nuevo usuario, como se muestra en la imagen 2.

Creando un nuevo usuario para la plataforma Israel Reyes 

Correo Perfil


Administrador 

Lo sentimos pero correo electronico es obligatorio.

Nombres Apellidos

El nombre es obligatorio. El apellido es obligatorio.

Empresa

Seleccione una opción 

La división es obligatoria.

Contraseña

La contraseña es obligatorio.

Observacion

Guardar ← Volver

Imagen: Ventana Creación de Usuario

En dicho mantenedor, se consideran los siguientes campos a completar por parte del encargado de registrar al nuevo usuario:

- Correo electrónico.
- Nombre
- Apellido
- Empresa
- Contraseña
- Perfil de usuario, del cual se pueden ingresar los siguientes roles: Super Administrador, Administrador, Operador.

Desarrollo del Caso de Uso

Como se pudo observar, dentro de la página de ingreso de usuario, se nos muestran varios campos dentro del formulario, se debe indicar que cada campo posee las validaciones respectivas, como lo son el validar que el correo electrónico sea efectivamente uno, como también el que, al momento de ingresar al usuario, el sistema indique que el usuario ya existe.


Las pruebas a estos mantenedores tanto de ingreso, edición y eliminación fueron realizadas durante el desarrollo de la aplicación, en conjunto con la empresa Arauco, con el fin de corroborar que la información ingresada sea validada e ingresada a la base de datos.

Una vez realizado esto, el sistema fue puesto en un servidor de pruebas para que personal asignado de Arauco pudiera entregar feedback al respecto, con la finalidad de poder realizar cambios y mejorar.

Este proceso fue llevado a cabo en todos los mantenedores del sistema y una vez completado dicho proceso se lanzó un prototipo, el cual fue enviado al cliente para pruebas y revisión.

Una vez que el cliente entregó su feedback se procedió a corregir lo necesario, para finalmente lanzar la versión final para que el cliente hiciera uso de la plataforma.

En la siguiente imagen, vemos la pestaña para editar un usuario, donde se aprecia que tanto el correo (clave primaria) no es editable, pero si los campos de nombre, apellido, empresa, contraseña, observación, registro activo (deshabilitar o habilitar usuario del sistema) y perfil.

Modificando usuario de la plataforma Israel Reyes 

Correo: alexis.aguayo@arauco.com Perfil: Administrador

Nombres: Alexis Apellidos: Aguayo Toro

Empresa: Arauco Viveros

Contraseña: *****

Observacion: Administrador de botones de alerta para Viveros en Horcones

Registro activo

[Guardar](#) [← Volver](#)

Imagen: Edición de Usuario

Así mismo, en la siguiente se puede observar la pestaña de Eliminación de un Usuario, donde pueden ocurrir dos casos.

Caso n°1

Si el usuario no está vinculado a otra tabla del sistema, no se podrá deshabilitar, debido a que puede afectar los datos históricos, y reportes futuros, en este caso, el sistema solo pone al usuario inactivo, para poder activar se deberá seleccionar el usuario inactivo desde la vista principal y luego en la vista editar se pondrá como activo.

Caso n°2

En el caso de que el usuario no tenga relación con las tablas del sistema este será borrado para siempre.

Eliminando Usuario de la plataforma Israel Reyes

Correo: adminViveros@arauco.com Tipo usuario: Administrador

Nombre completo: Admin Viveros

Division: Arauco Viveros

Registro activo


[Eliminar](#) [Volver](#)

Imagen: Eliminación de Usuario

Estos métodos de pruebas y error fueron realizados de forma similar en todos los mantenedores del sistema, y una vez aprobados por el cliente fueron puestos en producción.

Pruebas de Caja Negra

Dentro de la Prueba de Caja negra, se observó que los resultados obtenidos al momento de ingresar un nuevo usuario, fueran reflejados tanto en la aplicación como en la base de datos correspondiente.

Modificando usuario de la plataforma Israel Reyes 

Correo: Perfil:

Nombres: Apellidos:

Empresa:

Contraseña:

Observación:

Registro activo

Imagen Creación de Usuario

Una vez ingresado los datos, estos son procesados internamente a través de los métodos correspondientes, dando como resultado el ingreso del nuevo usuario a la aplicación, el cual se ve reflejado tanto en el sistema como en la base de datos respectiva, como se observa en la siguiente imagen.



Imagen: Nuevo Usuario

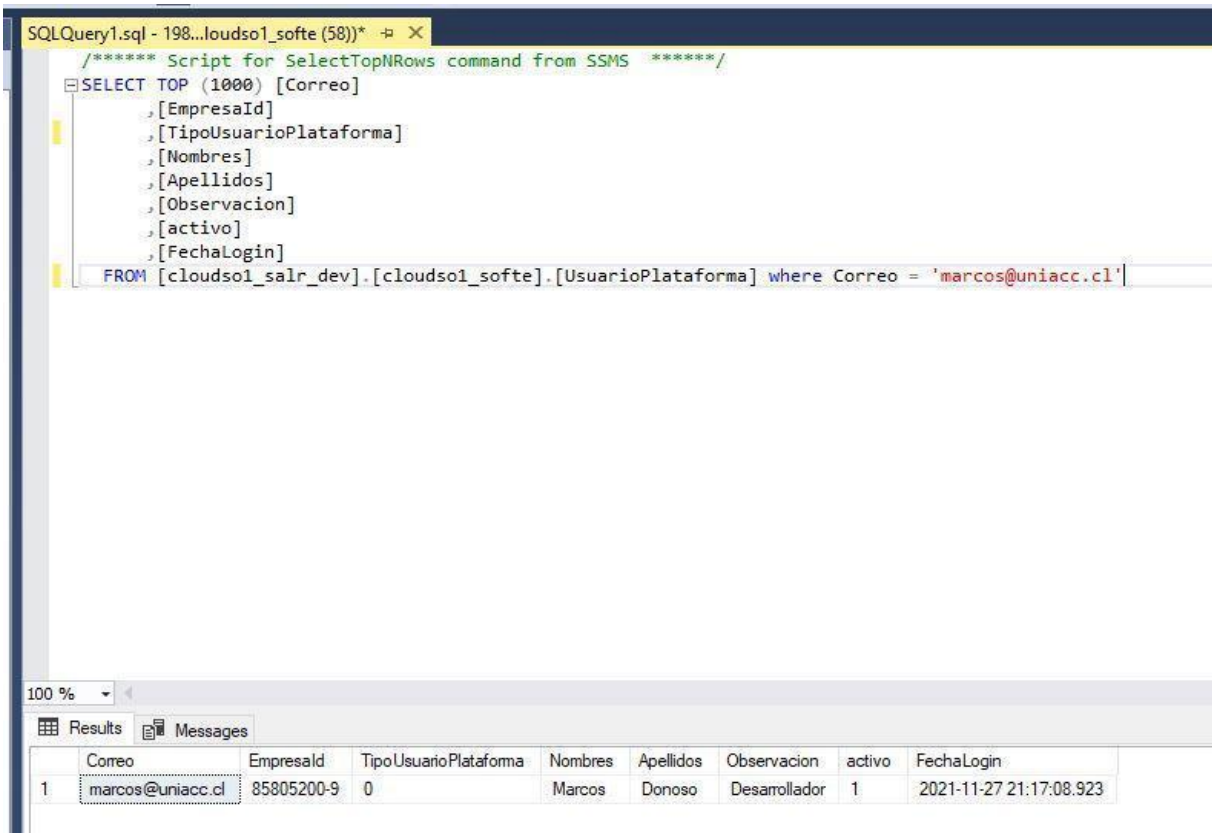


Imagen: Nuevo Usuario en Base de Datos

Como resultado final de la prueba, se pudo obtener que el ingreso correspondiente, fue exitoso y que cumple con lo esperado por parte del cliente, así mismo indicar que el usuario posee todos los datos que son solicitados tanto en la Base de datos como dentro del perfil de este.

Pruebas de Caja blanca

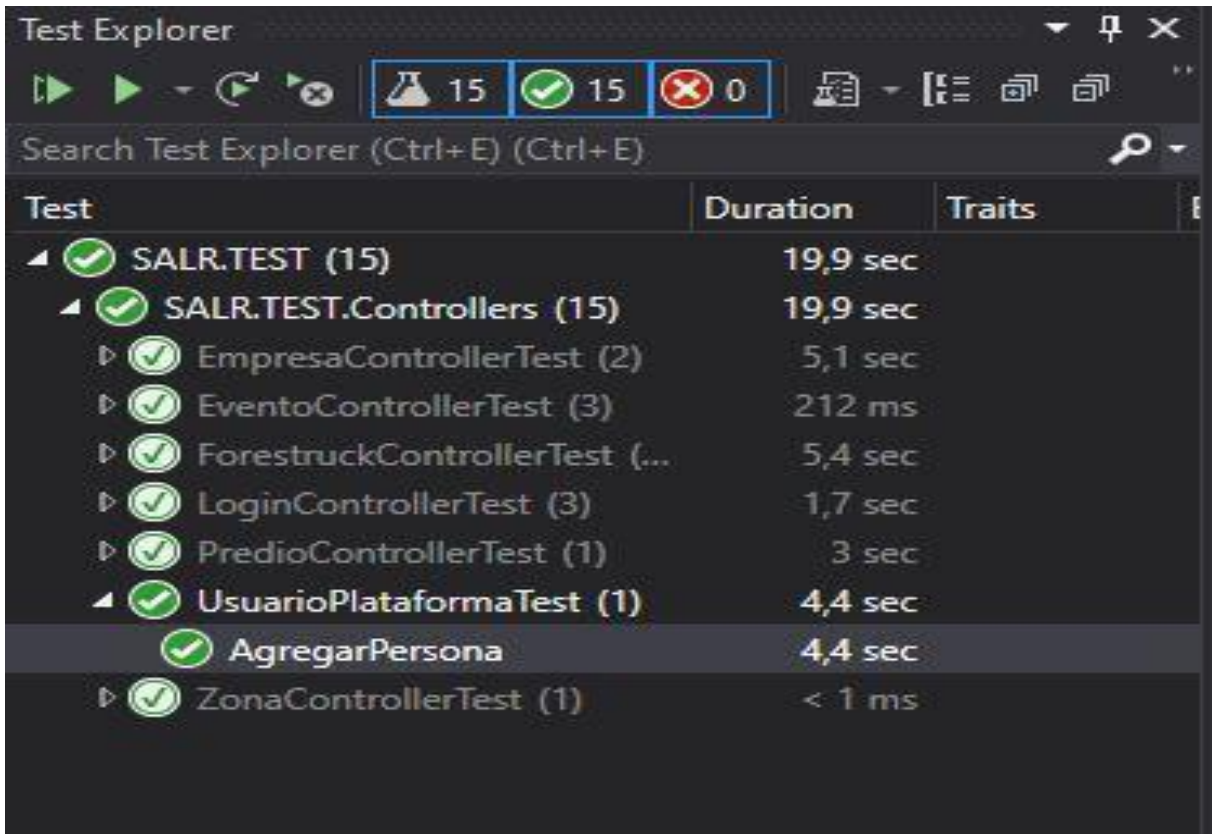
Se observará cómo funciona internamente el método para ingresar a un usuario el cual tiene como nombre AgregarPersona, donde se procede a llamar al controlador correspondiente y utilizar la clase UsuarioPlataforma.

```
[TestMethod]
✓ | 0 references
public void AgregarPersona()
{
    UsuarioPlataformaController controller = new UsuarioPlataformaController();
    SALR.EF.UsuarioPlataforma data = new SALR.EF.UsuarioPlataforma
    {
        Correo = "gb@gmail.com",
        Nombres = "Gabriel",
        Apellidos = "Boric",
        EmpresaId = "76110809-3",
        TipoUsuarioPlataforma = 0,
        Clave = "123test123$",
        Observacion = "",
        activo = true
    };
    var result = controller.Create(data);

    Assert.IsNotNull(result);
}
```

Imagen: Método Ingreso de Usuarios

Una vez llenando los datos estáticos con la información correspondiente, se ejecuta la prueba respectiva.



The screenshot shows the Test Explorer window in Visual Studio. At the top, there are icons for running tests: a green play button, a refresh button, a stop button, and a summary bar showing 15 tests passed (green checkmark), 15 tests passed (green checkmark), and 0 tests failed (red X). Below the icons is a search bar. The main area is a table with columns for Test, Duration, and Traits. The test 'AgregaPersona' is highlighted in blue.

Test	Duration	Traits
✓ SALR.TEST (15)	19,9 sec	
✓ SALR.TEST.Controllers (15)	19,9 sec	
✓ EmpresaControllerTest (2)	5,1 sec	
✓ EventoControllerTest (3)	212 ms	
✓ ForestruckControllerTest (...)	5,4 sec	
✓ LoginControllerTest (3)	1,7 sec	
✓ PredioControllerTest (1)	3 sec	
✓ UsuarioPlataformaTest (1)	4,4 sec	
✓ AgregarPersona	4,4 sec	
✓ ZonaControllerTest (1)	< 1 ms	

Imagen: Prueba Creación de Usuario

Una vez ejecutado el procedimiento, este es revisado y como se observa en la siguiente imagen, demora aproximadamente 4,4 segundos en ser ejecutado.

Observaciones sobre los resultados obtenidos

Como el sistema se encuentra en producción actualmente, este ya pasó por distintas pruebas durante su desarrollo, siendo evaluado positivamente por el cliente, teniendo pocas observaciones finales a la hora de ser entregado.

Lo anterior debido a la participación de la misma empresa Arauco en las pruebas de desarrollo, con el fin de mejorar el comportamiento del sistema con respecto a los requerimientos establecidos y su funcionalidad respectiva.

Si bien en lo que respecta a nuevas pruebas, ya sean de explotación, las cuales tratan de estresar los servidores y también de redundancia, aún no se han recibido observaciones en producción, más allá de temas indicados por los usuarios, que generalmente resultan ser problemas del tipo operativo.

Conclusión

En virtud de lo expuesto en los capítulos anteriores, es posible concluir que la creación y ejecución del sistema antes descrito cumple a cabalidad con las necesidades señaladas por el usuario final del producto, sin embargo, es un sistema perfectible que puede ir ajustándose, en el caso de ser necesario, a nuevos requerimientos por parte del contratante.

A pesar de lo anterior, actualmente las bases del sistema permiten que este sea un modelo de negocios robusto y escalable el cual permite ser aplicado y ejecutado por cualquier empresa o usuario que requiera implementar un sistema de alerta.

Recomendaciones

- En un proyecto de software como este se recomienda el uso de una metodología como la SCRUM, ya que tiene muchas ventajas prácticas frente a otras.
- La mejora continua es clave para el éxito a largo plazo.
- La entrega de un MVP (Minimum Viable Product) temprano en el desarrollo da la ventaja de poder entregar al cliente una versión funcional para evaluación y pruebas, lo que permite poder fallar temprano en caso de que los clientes no estén satisfechos.

Referencias

Solé, M. (2019) Pruebas de Caja blanca y Caja Negra. Apunte de clase unidad 1, Proyecto de Integración IV, Universidad UNIACC.

Solís, O. (2017). Pruebas y evaluación del cliente. Comercio Electrónico y Webengineering. Lea esto primero (Semana 5).

Team, D. (2021, May 12). What are the Features of Selenium IDE? DataFlair. Retrieved December 28, 2021, from <https://data-flair.training/blogs/selenium-ide-features/>

Solé, M. (2019) herramienta de pruebas en la web. Apunte de clase unidad 3, Proyecto de Integración IV, Universidad UNIACC.

El Mercurio de Santiago. (junio 25, 2012). Forestal Arauco se convierte en mayor propietario de bosques de América Latina. abril 13, 2022, de Mundo Marítimo Sitio web: <https://www.mundomaritimo.cl/noticias/forestal-arauco-se-convierte-en-mayor-propietario-de-bosques-de-america-latina>

Administrador. (septiembre 03, 2021). Ataques incendiarios contra Forestal Arauco y empresa de áridos dejaron múltiples maquinarias destruidas. abril 13, 2022, de Resumen Sitio web: <https://resumen.cl/articulos/ataques-incendiarios-contra-forestal-arauco-y-empresa-de-aridos-dejaron-multiples-maquinarias-destruidas>