

INSTITUTO PROFESIONAL DE CHILE

**“PREVCRIM Sistema Integrado de registro para la prevención”**

NOMBRE ESTUDIANTE(S): Sebastián Vargas

Gastón Velásquez

CARRERA:

Analista programador computacional

SEDE: Republica

PROFESOR DEL MÓDULO:

José Cristian Andrés Arellano Castillo

**REPÚBLICA, JULIO 2020**

*Índice.*

* Resumen Ejecutivo Página 3 -- 5.
* Introducción Página 6 -- 7.
* Antecedentes Generales Página 8.
* Diagnóstico Página 9 -- 10.
* Objetivos Página 11 – 12.
* Metodología & Planificación Página 13 – 24.
* Ejecución & Control de Proyecto Página 25 – 36.
* Conclusión & Reflexiones Página 37 – 39.
* Bibliografía Página 40.
* Anexos Página 41 – 50..

*Resumen Ejecutivo.*

La inseguridad en la población aumenta conforme pasan los días, llegando niveles inaceptables con respecto a la calidad de vida de la población. Por muchos intentos que las autoridades concentren, a fin de cuentas, estos esfuerzos no han reflejado una mejoría en lo que respecta a esta problemática.

La organización que se dedica a prevenir delitos y mejorar la seguridad de la población, PREVCRIM, nos presenta sus avances con respecto a la problemática. Específicamente esta empresa desarrolla y comercializa aplicaciones para prevenir delitos. Este servicio (desarrollo de software) está pensado para autoridades ya sea privadas o estatales, permite el registro de crímenes, criminales y delitos con el fin de poder generar informes para prevenir estos percances.

Se propone la contratación para el desarrollo de un “Sistema Integrado de Registro para la prevención” (SIRP), su propósito es administrar datos personales de los delincuentes como delitos, lugar de los delitos, identificación del delincuente.

Actualmente sí se cuenta con un sistema de registro (AFIS: Sistema de identificación de huella dactilar) este sistema accede al servicio “Monito Web” y que en caso de verificar una identidad por sistema biométrico utiliza el software SECUGEN. El problema con estos sistemas es que son independientes entre sí, solo sirven para consultas, y las unidades policiales llevan su propio registro de incidentes por localidad.

Entre los requerimientos que se presentan para desarrollar un nuevo software se nos solicita identificar usuarios en el sistema. Esto debe seleccionar entre entidades ya sea Carabineros, paz ciudadana, u otras autoridades. Deben estar autorizadas por el administrador y no se podrá manipular los datos, solo agregar datos y generar reportes (solo en el caso del jefe de zona). Los usuarios deben iniciar sesión con sus credenciales (nombre, institución, contraseña) y su perfil (Administrador General, Jefe de Zona, Operador).

Además de esto último, nuestro sistema debe generar informes con listados por ejemplo de:

* Delincuentes por delito cometido.
* Delincuentes por Comuna o sector.
* Rankin de comunas con mayor cantidad de delitos.

Además, el sistema debe validar, datos de delincuentes ingresados, opciones de ingreso de datos, validar campos en interfaz gráfica, validar campos numéricos y todos los errores que vayan sucediendo durante la ejecución deben ser señalados claramente.

*Executive Summary.*

Insecurity in the population increases as the days go by, reaching unacceptable levels with respect to the quality of life of the population. Although the authorities have made many attempts to concentrate these efforts, in the end they have not led to an improvement in this problem.

The organization dedicated to preventing crime and improving the security of the population, PREVCRIM, presents its progress with respect to the problem. Specifically, this company develops and markets applications to prevent crime. This service (software development) is thought for authorities, either private or state, it allows the registration of crimes, offenders and offences in order to be able to generate reports to prevent these mishaps.

It is proposed to contract for the development of an "Integrated Registration System for Prevention" (SIRP), its purpose is to manage personal data of offenders as crimes, place of crimes, identification of the offender.

Currently there is a registration system (AFIS: Fingerprint Identification System) this system accesses the service "Monito Web" and in case of verifying an identity by biometric system uses the SECUGEN software. The problem with these systems is that they are independent from each other, they only serve for consultations, and the police units keep their own record of incidents by location.

Among the requirements that are presented to develop a new software we are asked to identify users in the system. This must be selected among entities whether they are Carabineros, Paz Ciudadana, or other authorities. They must be authorized by the administrator and cannot manipulate the data, only add data and generate reports (only in the case of the zone chief). Users must log in with their credentials (name, institution, password) and their profile (General Administrator, Zone Chief, Operator).

In addition to this last one, our system must generate reports with listings for example of:

- Offenders by crime committed.

- Offenders by Community or sector.

- Ranking of communes with the highest number of crimes.

In addition, the system must validate, data of entered criminals, options of data entry, validate fields in graphic interface, validate numerical fields and all the errors that are happening during the execution must be indicated clearly.

*Introducción.*

Es un hecho que la seguridad a nivel nacional no es óptima en lo que respecta a los últimos años. Por uno u otro motivo las soluciones propuestas no han ayudado disminuir los delitos que se cometen día a día.

Antes de hacer un pequeño desglose de lo que contiene nuestro proyecto debemos mencionar las consecuencias de la mala prevención en el ámbito de la seguridad. No es descabellado decir que somos más propensos a un “portonazo” que sufrir alguna enfermedad, la seguridad es un tema de primera importancia. Nuestro proyecto pretende ser una gran ayuda a las autoridades correspondientes, uno de nuestros varios objetivos es ser un aporte a los ciudadanos a través de nuestros servicios.

Contemplamos el hecho de que los sistemas que hay son independientes entre sí, y nuestro propósito como tal ayudar a administrar de forma más eficiente la información relacionada a los delincuentes, como, por ejemplo: delitos del imputado, lugar en donde se cometieron los delitos.

Dentro de los requisitos se nos pide identificar a los usuarios del sistema. Estos van tres perfiles de usuarios que se describen a continuación.

* Administrador General: Posee a todas las funcionalidades del sistema que son: creación, actualización y eliminación de instituciones y usuarios.
* Jefe de Zona: Puede crear nuevos usuarios en una determinada institución, crear sectores para el monitoreo y generación de reportes.
* Operador: Ingresa la información de los delincuentes, delitos, controles realizados, antecedentes.

Los reportes deben generar los siguientes listados, que pueden ser visualizados en papel impreso o pantallas en algún dispositivo:

* Delincuentes por orden alfabético.
* Delincuentes por delito cometido.
* Delincuentes por comuna de residencia.
* Última comuna donde se vio a un determinado Delincuente.
* Parentesco entre determinados Delincuentes.
* Delitos ocurridos en una fecha específica por Comuna donde sucedió éste.
* Cantidad Delitos ocurridos por sector específico.

Además de funciones como:

* Búsqueda por cualquier campo.
* Ranking de comunas con mayores delitos dependiendo de una fecha.

Con respecto a las credenciales del usuario antes de entrar al sistema, son las siguientes:

* Nombre de Usuario.
* Contraseña.
* Perfil de Usuario.
* Institución a la que pertenece el Usuario (Paz Ciudadana, Seguridad OS10, PDI, entre algunos ejemplos).

\*Estas instituciones deben estar autorizadas por algún ente que tenga perfil de usuario correspondiente a Administrador General\*.

Nuestro proyecto validará por lo demás: rut del delincuente (no debe estar duplicado al ingresar), validar los datos ingresados en la interfaz gráfica, validar que en los campos donde se deben ingresar datos de tipo numérico no se ingresen letras, restringir el largo de un dato ingresado.

Antecedentes Generales.

Se solicita la creación de un Sistema Integrado de Registro para la Prevención, este sistema lo solicita la empresa PREVCRIM, empresa que se dedica a desarrollar y comercializar diferentes tipos de aplicaciones que sirven para la prevención de delitos.

Este Sistemas debe estar dirigido a las policías ya sea privadas, como estatales. El fin de este Sistema debe ser obtener informes para la prevención de delitos, como también identificar delincuentes, además de funciones como:

* Recolectar información relacionada con los datos personales de los Delincuentes.
* Lugar donde se cometieron delitos.

Hay que mencionar que ya existen Sistemas para combatir esta problemática, el inconveniente con estos Sistemas de registro, pero no hay un Sistema Integrado de Información como tal. Es por ello que las autoridades deben consultar al sistema AFIS (Sistema de Huella Dactilar) por ejemplo, para acceder a órdenes de detención pendientes deben consultar un servidor “Monito Web”, y si necesitan verificar la identidad en un sistema biométrico deben consultar el software SECUGEN.

El gran problema con estos Sistemas es que son independientes entre sí, y que estos solo permiten a los Usuarios son consultar información, están limitados a que cada unidad policial debe contar con su propio registro de Incidentes dependiendo de su localidad.

*Diagnóstico.*

Se nos presenta una problemática bastante interesante, como mencionamos anteriormente nuestra perspectiva en lo que confiere a la seguridad es bastante desalentadora, la población en general tiende a tener miedo al salir a la calle.

La cuestión a solventar solo depende de cómo llevemos cabo el proyecto, proyecto que consiste en desarrollar un Sistema Integrado de Registro para la Prevención.

Nuestra solución consiste en un software desarrollado bajo el lenguaje de programación Java, combinado con el motor de base de datos MySQL. Este software se administrará vía Web, esto para facilitar el manejo de la información entre instituciones pertinentes.

En lo que respecta a nuestros detractores a evitar debemos mencionar que los sistemas existentes que están disponibles en la actualidad presentan las siguientes inconsistencias, las cuales debemos solventar a través de nuestro software:

* Son Sistemas Independientes unos de otros.
* No hay un entrelazado, entrecruzado de datos entre sistemas.
* Existe una carencia en lo que respecta a herramientas de análisis que se puedan usar para prevenir un delito.
* Hay pérdida de información en estos sistemas.
* No hay un acceso satisfactorio para ingresar al sistema por instituciones para acceder a información relevante dependiendo de la institución.
* No hay un diseño integrado de software óptimo que use toda la información pertinente.
* Existen inconvenientes a la hora de buscar a través de la planilla de Excel, se gasta tiempo demás

Nuestro Software debe tener la funcionalidad de Identificar los usuarios del sistema (Administrado General, Jefe de Zona, Operador), generar Reportes, validar ciertos registros antes de ingresarlos en la base de datos que corresponda.

Debemos mencionar que nuestro software debe mantener la información de los delincuentes con volumen de 5000 transacciones mensuales. Esta base de datos debe tener tablas modelar entidades como: Delincuentes, Delitos, Operadores, Instituciones, Sectores y Usuarios. El modelo de datos lo explicaremos más avanzado este informe, todas sus características y cómo es que funciona este modelo dentro de nuestro proyecto.

La Interfaz gráfica de nuestro proyecto debe ser sencillo y intuitivo de usar. Esta interfaz debe tendrá el nombre de la empresa que solicita nuestros servicios, la empresa “PREVCRIM” que no debe ocupar más del 10% del formulario. Esta interfaz debe enlazarse con los distintos JFrame y páginas Web en las aplicaciones Web.

Por último y no menos importante el ingreso al sistema debe estar ligado a dos atributos importantes, Nombre de Usuario (Creado por el administrador General) y Contraseña (Creadas- por el administrador General. Estas contraseñas deben estar encriptadas en la base de Datos). Cabe mencionar que los usuarios del sistema no podrán tener acceso a cuentas de otros usuarios del sistema.

Login


*Objetivos.*

Uno de los procesos elementales a la hora de realizar el trabajo, fue el análisis de los requerimientos, el enunciado como tal que se nos presentó, leer todos los antecedentes que logramos entender. Luego de este análisis, se realizó una lista con todos los requerimientos solicitados, estos a posterior se fueron clasificando por hitos. Una vez separados los requerimientos por hitos, se comenzó el desarrollo del software por versiones (Prototipos del software que explicaremos más adelante).

Como primer contacto se planteó hacer un software que cumpliese con todos los aspectos de la rúbrica que se nos asignó lógicamente. Desde un principio se planteó realizar el software en Java con el fin de familiarizarse aún más con este lenguaje de programación.

Se plantea hacer un software de enterado de registros para la prevención de crímenes. Este Software registra todos los crímenes, y combinado con la base de datos se pueden generar reportes / informes.

Debemos confirmar que, finalmente el software se decidió hacer en Java con la idea que mencionamos anteriormente, el de familiarizarse aún más con el ambiente de la mano con MySQL.

Nuestro Objetivo Específico fue lograr realizar el software bajo el pensamiento de establecer algunas ideas para establecer qué es lo que necesitamos, qué debemos condicionar dentro de nuestro software, qué decisiones debemos tomar con respecto al formato de nuestro trabajo, qué es importante a la hora de atender primero un criterio por sobre otro, cuáles son nuestros requerimientos clave, en qué no debemos equivocarnos, y por último, cuáles son los aspectos que no debemos subestimar dentro de este proyecto.

Debido a que hay que unificar todos los datos, todos los datos irán a una base de datos en común. Y que como había problemas al ingresar datos, también había pérdida datos.

Como se mencionó anteriormente, para trabajar como primer paso se levantaron los requerimientos, a posterior se comenzó a programar todo lo que tiene que ver con la interfaz gráfica, es decir: Diseños del software, la fluidez del software, y pensar como es que el usuario va a interactuar con el programa.

Como primer paso dentro del pensamiento para programar, se tomó en cuenta el acceso, el inicio de sesión del usuario y qué se necesita para lograr el acceso a la aplicación. Posterior a esto se pensó en el menú de opciones. Durante este proceso consideramos elemental la identificación del usuario en el sistema, es decir su perfil de usuario en este orden: Administrador General (¿A qué tiene acceso?, dependiendo de ese cuestionamiento se procede a programar), Jefe de Zona (¿A qué tiene acceso?, dependiendo de ese cuestionamiento se procede a programar. Al tener opciones similares a las de un Administrador General se programó que el Sistema reconozca quién, qué tipo de usuario, está haciendo la consulta), Operador (El mismo criterio que los anteriores tipos de usuarios). En resumen, se generaron filtros para determinar qué acciones puede hacer cada tipo de usuario y qué opciones se le mostrarán al usuario en la interfaz gráfica, y también qué funciones no podrá realizar.

¿Cuál es la finalidad de nuestro trabajo?

La finalidad / Objetivo General de nuestro trabajo es crear un software que tenga las siguientes características, que también se mencionaron en el diagnóstico:

* Exista un entrelazado de datos entre sistemas.
* No falten herramientas a la hora de prevenir un delito.
* No se pierda tiempo al buscar cierto tipo de información.
* Haya un diseño de software que use toda la información pertinente.
* Permita un acceso sencillo, intuitivo con fluidez al sistema.

*Metodología y Planificación.*

El formato de nuestro proyecto se resume en un software creado bajo el lenguaje de programación Java en la plataforma NetBeans, este software se apoya en un base de datos creada en MySQL a través del software XAMPP.

Para desarrollar todas las funciones necesarias y solventar uno a uno los requerimientos nos apoyamos de las siguientes librerías:

* Placeholder:

Permite visualizar un texto, indicar que debe ingresar un usuario en un determinado formulario, luego de ingresar algún dato este texto desaparece

* MYSQL – JDBC:

Librería que permite la conexión de nuestro programa con la base de datos en MySQL.

* Commons-codec-1.14:

Librería que permite encriptar las contraseñas en la base de datos con MD5.

* JXL:

Librería que permite crear archivos Excel.

* JCalendar:

Librería que permite crear Calendarios en Java.

* JFreeCharts:

Librería que permite generar gráficos en java.

* IText:

Librería que permite generar archivos PDF.

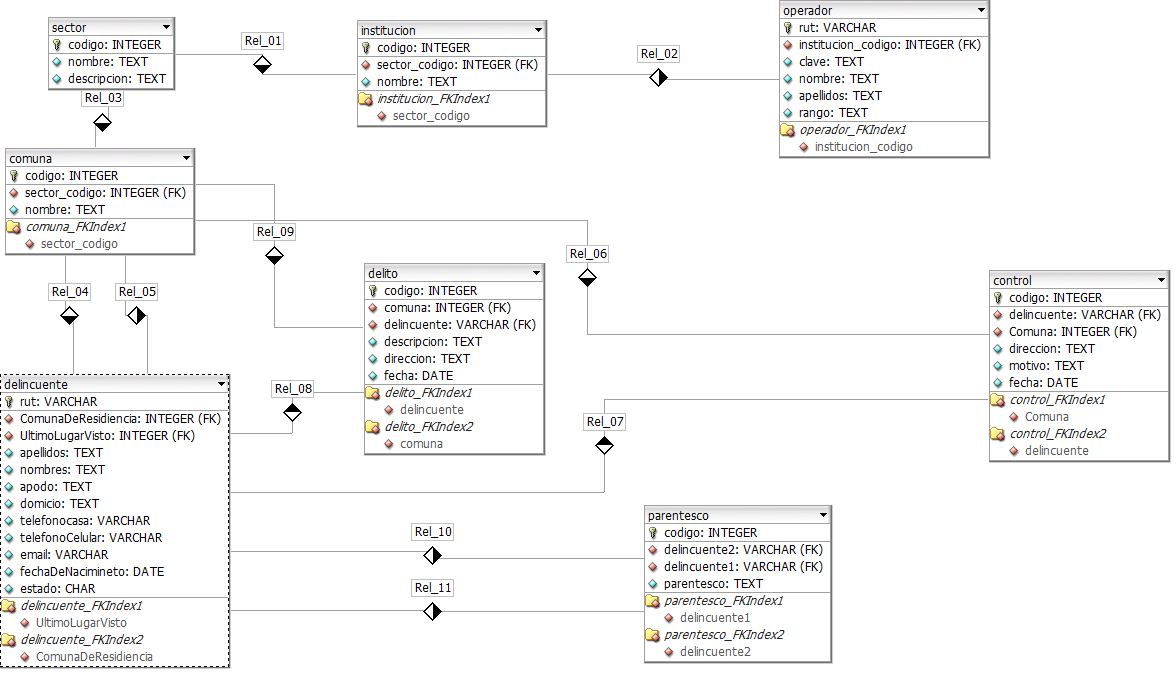
El problema que resalta se presenta en la base de datos donde en los otros sistemas que se nos mencionan no hay orden en los datos, los datos van a diferentes lugares donde no deberían ir. Nuestra solución consta en agrupar todos los datos en una base de datos en común, a una sola base de datos.

Nuestro sistema permite el ingreso de distintos usuarios (distintos tipos de Usuario) de distintas instituciones. Nuestro sistema permite que un usuario de un determinado sector tiene acceso a los datos de su sector correspondiente, sólo a ese sector; por lo tanto, este usuario sólo podrá realizar acciones pertinentes dentro de ese sector. Cabe mencionar que esto último depende de la institución a la que pertenezca este usuario, debido a que la institución depende del sector ya que puede haber varias instituciones en un mismo sector.

El sistema desarrollado también crea estadísticas, además de reportes en formato PDF y en formato Excel. El formato en PDF sirve en los casos en que este reporte esté impreso y necesite la firma de alguna autoridad. En cambio, los reportes en formato Excel son útiles en caso de que el trabajo asignado necesite un análisis de datos.

Para guiarnos durante la realización del proyecto pensamos en el desglose de los requerimientos es por ello que partimos programando la pantalla de inicio de sesión, a posterior se pensó en desarrollar un “SplashScreen” que mientras carga esta imagen su función es comprobar que el programa en Java esté conectado a la base de datos, o que haya algún imperfecto o inconveniente con la base de datos, una vez verificado la conexión o algún inconveniente, el programa mostrará la pantalla de inicio de sesión.

*Modelo de Base de Datos.*



Se utilizó XAMPP, más específicamente MySQL plataformas que pueden ser reemplazadas cuando el modelo se hace ya implementado dentro del software, pueden ser reemplazados por un servidor LAMP en Ubuntu / Linux y el programa en Java solamente se instala en el dispositivo, este se conectará a la base de datos a través del servidor.

Según los requerimientos del sistema se nos pidió que debían estar presente: instituciones, operadores, sectores, registro de delitos, registro de delincuentes y parentesco entre delincuentes.

Para entender un poco mejor nuestro modelo de base da datos vamos a explicarlo brevemente a continuación:

* Un operador debe pertenecer a una institución.
* Una institución deber pertenecer a un sector, puede que varias o muchas instituciones pertenezcan a un solo sector.
* Un sector debe pertenecer a una comuna, puede que varios o muchos sectores pertenezcan a una sola comuna.
* Un delincuente debe pertenecer a una comuna.
* Un delincuente tiene un atributo que es “Último lugar visto” que va relacionado también con una comuna.
* El parentesco va relacionado entre dos o más delincuentes. Un delincuente puede tener parentesco con muchos otros delincuentes.
* Los controles van relacionados con una comuna y con un delincuente. Un delincuente puede tener muchos controles.
* Nuestro sistema tiene la particularidad de que cuando se realiza un control nuevo a un delincuente, el atributo dentro de un delincuente “último lugar visto” se actualiza con la comuna donde se realizó el control respectivamente.
* Un delincuente / comuna puede tener muchos delitos.

*Carta Gantt.*













*Casos de Uso.*

|  |  |
| --- | --- |
| Nombre | Logear |
| Autor | Sebastian Vargas |
| Actor | Usuario |
| Descripción | Proceso necesario por el cual el usuario ingresa al sistema PREVCRIM. |
| Disparador | Boton en la login screen “Entrar” |
| Flujo básico | 1. Ingresar datos en el formulario de Ingreso(“Login”)  2. Verificar rut valido.  3. Verificar Datos ingresados  4. Verificar tipo de Usuario  5. pasar a menu principal |
| Flujos alternos | 2.1. En caso de ingresar rut no valido, mostrar mensaje e ir a a 1  3.1 Si los datos no son ingresados, mostrar mensaje e ir al paso 1 |
| Pre-condiciones | 1. Cliente debe estar registrado como Usuario |
| Post-condiciones | Pasar al proceso de carga de menu. |

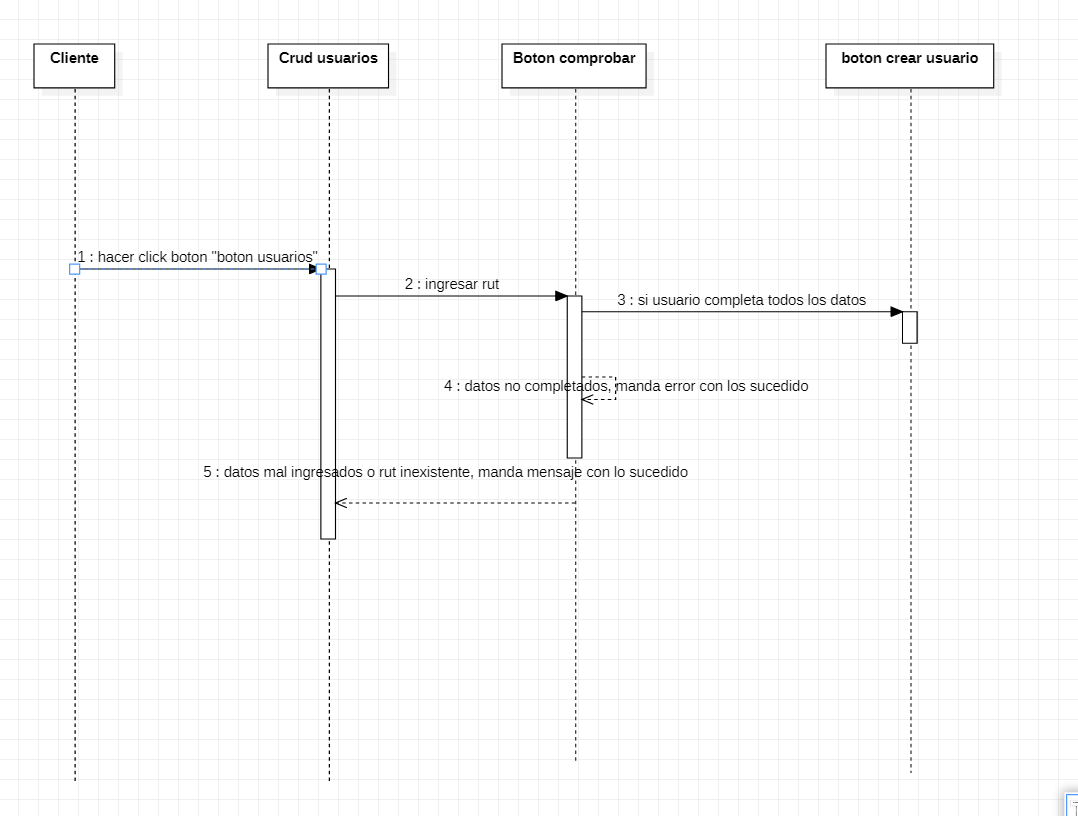
|  |  |
| --- | --- |
| Nombre | Cargar menu |
| Autor | Sebastian Vargas |
| Actor | Usuario |
| Descripción | Proceso por el cual el sistema identifica y muestra el menu dependiendo, que rango tiene el usuario que interactua con el sistema . |
| Disparador | Login correcto |
| Flujo básico | 1. El sistema recibe los datos del usuario(Nombre, sector, rut, rango) 2. El sistema guarda que rango tiene el usuario 3. El sistema compara el rango y activa botones dependiendo que rango tiene el usuario |
| Flujos alternos |  |
| Pre-condiciones | 1. El usuario debe haber pasado por la pantalla de login |
| Post-condiciones | Mostrar menu de usuario |

|  |  |
| --- | --- |
| Nombre | Crear un usuario |
| Autor | Sebastian Vargas |
| Actor | Jefe de zona, Administrador general |
| Descripción | Proceso por el cual se procede a crear un usuario dentro del sistema . |
| Disparador | Botono Usarios |
| Flujo básico | 1. El usuario ingresa un rut a el campo “Rut”, 2. El usuario apreta el boton comprobar. 3. El sistema comprueba si el campo contiene datos. 4. El sistema comprueba si el usuario no existe. 5. El sistema activa los campos para que usuario introduzca los datos. 6. El usuario ingresa todos los campos requeridos y presion el boton “crear”. 7. El sistema comprueba que esten los campos completos. 8. El sistema crea el nuevo usuario. |
| Flujos alternos | 3.1 El usuario no ingreso datos y el sistema manda un mensaje con el error y el regresa al punto1  4.1 El rut ya esta registrado y lanza un mensaje que le informa al usuario de esto. El sistema habilita al opcion de editar y eliminar . Se regresa al punto 1  7.1 El usuario no completa todos los datos y el sistema manda un mensaje informando lo ocurrido, de regreso al punto 6 |
| Pre-condiciones | 1. El usuario debe haber cargado el menu |
| Post-condiciones | Se crea un usuario a la base de datos |

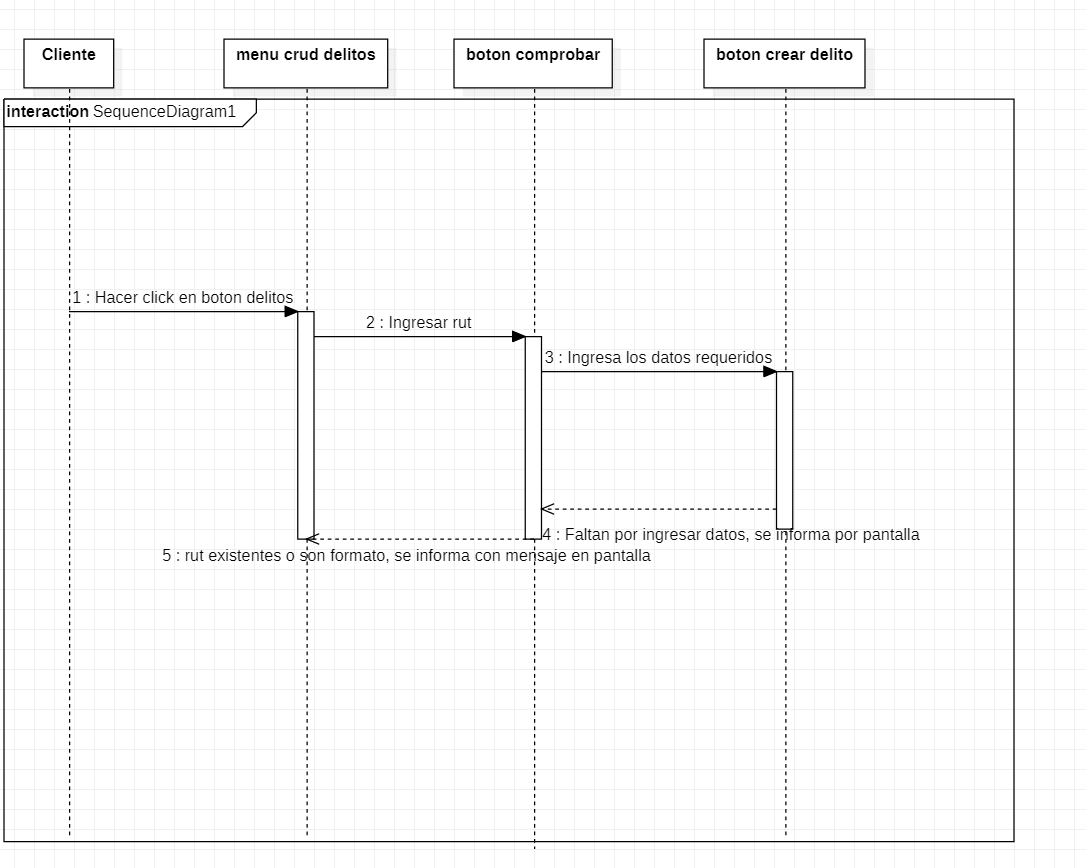
|  |  |
| --- | --- |
| Nombre | Eliminar un usuario |
| Autor | Sebastian Vargas |
| Actor | Jefe de zona, Administrador general |
| Descripción | Proceso por el cual se procede a eliminar un usuario dentro del sistema . |
| Disparador | Botono Usarios |
| Flujo básico | 1. El usuario ingresa un rut a el campo “Rut”, 2. El usuario apreta el boton comprobar. 3. El sistema comprueba si el campo contiene datos. 4. El sistema comprueba si el usuario no existe. 5. El sistema activa los campos para que usuario edite o elimine el usuario. 6. Usuario presiona boton “Eliminar” 7. El sistema elimina el usuario. |
| Flujos alternos | 3.1 El usuario no ingreso datos y el sistema manda un mensaje con el error y el regresa al punto1  4.1 El rut no esta registrado y lanza un mensaje que le informa al usuario de esto. El sistema habilita al opcion crear usuario . Se regresa al punto 1 |
| Pre-condiciones | 1. El usuario debe haber cargado el menu |
| Post-condiciones | Se elimina un usuario a la base de datos |
| Nombre | Ingresar datos de delito |
| Autor | Sebastian Vargas |
| Actor | Jefe de zona, Administrador general, Operador |
| Descripción | Proceso por el cual se procede a ingresar ingreso de un delito dentro del sistema . |
| Disparador | Botono delitos |
| Flujo básico | 1. El usuario ingresa un codigo a el campo “codigo”. 2. El usuario apreta el boton comprobar. 3. El sistema comprueba si el campo contiene datos. 4. El sistema comprueba si el delito existe. 5. El sistema activa los campos para que usuario introduzca los datos. 6. El usuario ingresa todos los campos requeridos y presion el boton “crear”. 7. El sistema comprueba que esten los campos completos. 8. El sistema crea ingresa el nuevo delito. |
| Flujos alternos | 3.1 El usuario no ingreso datos y el sistema manda un mensaje con el error y el regresa al punto1  4.1 El delito esta registrado y lanza un mensaje que le informa al usuario de esto. El sistema habilita al opcion editar o elimnar delito . Se regresa al punto 1  7.1 El usuario le faltan datos por ingresar y el sistema manda un mensaje de lo sucedido. De vuelta al paso 5 |
| Pre-condiciones | 1. El usuario debe haber cargado el menu |
| Post-condiciones | Se elimina un usuario a la base de datos |

*Diagramas de Secuencia*

* Diagrama de secuencia ingreso al sistema.



* Diagrama de secuencia ingreso de información de delitos al sistema.



*Problemas / Riesgos.*

* Riesgo1: Al ejecutar los Scripts de la base de datos, puede que se hayan ingresados datos erróneos, almacenando “datos basura” dentro de la base de datos.
* Riesgo2: El proyecto está almacenado en GitHub, pero no en su totalidad. Podría existir un inconveniente en caso de que el equipo de trabajo de un integrante dejé de funcionar correctamente, se podrían perder archivos y información vitalicia del proyecto.

*Metodología.*

El equipo de trabajo, realizo el proyecto bajo la metodología en cascada.

A medida que se fueron completando ciertos hitos, se comenzó a trabajar en otras áreas del proyecto, es por ello que esta metodología nos permitió comenzar áreas clave del proyecto con bastante tranquilidad al saber que las tareas previas para llegar hasta un determinado punto ya están completadas.

Las fechas indicadas en la planificación en la carta Gantt, nos permitieron trabajar con un tiempo estimado correcto entre una tarea y otra.

También debemos mencionar que las decisiones tomadas con respecto a aspectos importantes durante el desarrollo del software tuvieron que ser muy bien pensadas, debido a que la metodología es muy rígida con respecto a cambios imprevistos. Sin embargo, los requisitos propios del equipo de trabajo fueron solventados de buena forma y esto influyó en la fluidez del trabajo. No hubo rencillas entre integrantes del equipo, ni problemas mayores. Además, no se comenzó una tarea, sin haber terminado la tarea que le antecede a ésta.

A pesar de lo anterior mencionado con respecto a cambios inesperados, hubo muy pocos, y si solventaron de forma correcta y eficiente en su debido momento, el equipo de trabajo considera correcta la decisión de haber desarrollado el proyecto bajo la metodología en cascada.

*Ejecución y Control del Proyecto.*

*Prototipos.*

*Versión 0.01.*

Primer contacto del equipo de trabajo con los requerimientos solicitados, esta versión contenía o se compuso de la siguiente forma:

* Presencia del modelo de base de datos.
* Programación de la base de datos.
* Diseño de sistema de reporte.
* Contuvo la librería Placeholder (texto predictivo).
* Diseño de la página inicial, página de login.
* Programación de la página de inicio.
* SplashScreen.

*Versión 0.05.*

Durante el desarrollo de este prototipo se desarrollaron varias funciones adicionales y necesarias para resolver nuestros objetivos generales y nuestros objetivos específicos. Durante esta etapa el equipo de trabajo presentó bastantes planteamientos para lograr el objetivo en común, durante el desarrollo de este prototipo nos cuestionamos muchas posibles alternativas para tratar los requerimientos.

Este estuvo compuesto por las siguientes funciones:

* Programación de la página principal.
* Diseño de CRUD (Create, Read, Update and Delete) en la base de datos.
* Usuarios.
* Instituciones.
* Sectores.
* Comunas.
* Delincuentes.
* Parentesco entre delincuentes.
* Delitos.
* Controles.

Además de:

* Programación del sistema que genera los reportes.
* Mostrar delincuentes por orden alfabético.
* Listado de delincuentes agrupados por delito cometido.
* Mostrar delincuentes por comuna de Residencia.
* Mostrar delincuentes por último lugar visto.
* Mostrar delincuentes por parentesco entre ellos.
* Listado de delitos cometidos en una comuna o en un sector predeterminados por una fecha en específico.
* Listado histórico de delitos ocurridos en un sector en específico.
* Búsqueda de datos por un dato en particular.
* Ranking de comunas con mayor cantidad de delitos ocurridos dependiendo de fechas en específico.
* Ranking de sectores con mayor cantidad de delitos ocurridos dependiendo de fechas en específico.

En esta versión de nuestro sistema desarrollado faltaron funciones clave, estas fueron:

* No hay presencia de ningún tipo de validador de campos.
* Controles.
* Encriptación para Contraseñas.
* No hay presencia de gráficos, ni ningún tipo de estadísticas.

*Versión final.*

Finalmente se logró la creación y desarrollo del sistema que se nos presentó. Esta versión contiene todas las funciones mencionadas anteriormente y además funciona de forma correcta.

Cabe recalcar que está versión final será la que los profesores evaluarán.

Indicar esquemas y explicar esquemas.

¿Cuáles fueron las tareas para lograr un monitoreo exitoso?

Para lograr un monitoreo exitoso y que nuestro proyecto fuese desarrollado acorde con lo solicitado fue primordial guiarse de las bases técnicas de este proyecto. Cabe mencionar que para el equipo de trabajo no fue fácil interpretar algunas instrucciones, esto porque a pesar de que todo lo solicitado tenía fácil acceso, el razonar de la mejor manera, el análisis a la hora de desarrollar el software conforme a nuestras ideas fue un trabajo de mucho énfasis. Para entender este último punto debemos considerar que las ideas que tuvimos en su momento no se terminaban de alinear del todo con las bases técnicas del proyecto, en resumen, nuestras ideas tuvieron que encasillarse de alguna forma para poder encajar con los requisitos solicitados.

Para seguir adelante en el proyecto la manera que tomamos de pensar fue solventar uno a uno los requerimientos, no en un orden específico, sino que conforme fuese viable para nosotros el realizar una tarea. Es decir, si éramos capaces de realizar algún proceso entonces lo realizábamos, si no, nos concentramos en las tareas que sí podíamos realizar siempre y cuando estuviésemos cuadrados con respecto a los tiempos entre una actividad y otra. El hecho de haber trabajado de esta manera no fue un percance debido a que de una u otra forma las tareas se completaron de una u otra forma dentro de los tiempos establecidos.

Además de estos últimos dos puntos, fue no menos importante trabajar en el desarrollo del sistema siempre teniendo en cuenta las funciones a las que iba poder acceder cada tipo diferente de tipo de Usuarios, es decir, ¿A qué puede acceder?, en cambio, ¿A qué no puede acceder?, ¿Qué funciones son similares en un tipo de usuario con otro?, ¿Cuáles son los límites que debemos establecer en nuestro sistema?

¿Cómo fue que se llegó al objetivo general y específico?

Para lograr nuestro objetivo general que fue:

“La finalidad / Objetivo General de nuestro trabajo es crear un software que tenga las siguientes características, que también se mencionaron en el diagnóstico:

* Exista un entrelazado de datos entre sistemas.
* No falten herramientas a la hora de prevenir un delito.
* No se pierda tiempo al buscar cierto tipo de información.
* Haya un diseño de software que use toda la información pertinente.
* Permita un acceso sencillo, intuitivo con fluidez al sistema.”

Fue esencial la toma de decisiones, por ejemplo, el trabajar con Java desde un inicio desde un principio con todas sus ventajas y desventajas. Además de esto también después de analizar como debía modelarse la base de datos, se tomó en consideración agregar una tabla denominada “Controles”, “Parentesco (entre un delincuente y otro)” que a posterior nos ayudaría a manejar la información de mejor forma. Cabe resaltar que en breve explicaremos con mayor detalle el modelo de base de datos.

Para lograr que no se pierda tiempo buscando información en nuestro sistema, como sí pasa en los sistemas que se nos mencionó que existen; se agregó un campo “buscar (disponible en todas las tablas)” cuya función es filtrar todos los datos que haya en las filas de la base de datos bajo algún parámetro ingresado por el usuario.

Para lograr un acceso sencillo e intuitivo se investigó sobre colores agradables para usar en nuestra interfaz gráfica, además de ver diversas interfaces gráficas de otros proyectos con el fin de obtener ideas para lograr un diseño intuitivo y de fácil manejo para el usuario.

Se buscó primordialmente sencillez en este ámbito, realizar las funcionalidades bajo un criterio de accesibilidad y que fuese viable realizar para el equipo de trabajo, tener los conocimientos necesarios para lograr esa interfaz gráfica en la plataforma Java.

Para lograr nuestro objetivo específico se realizaron anotaciones sobre procesos dentro del proyecto, funcionalidades que debía tener el sistema. Posterior a anotar todo lo relevante y necesario, se comenzó a separar que funciones, procesos y características debían desarrollarse en primer lugar, con que debía se proceder y así sucesivamente hasta tener un plano general sobre las actividades a realizarse para lograr desarrollar todo.

*Casos de Uso Detallados.*

|  |  |
| --- | --- |
| Nombre | Logear |
| Autor | Sebastian Vargas |
| Actor | Usuario |
| Descripción | Proceso necesario por el cual el usuario ingresa al sistema PREVCRIM. |
| Disparador | Boton en la login screen “Entrar” |
| Flujo básico | 1. Ingresar datos en el formulario de Ingreso(“Login”)  2. Verificar rut valido.  3. Verificar Datos ingresados  4. Verificar tipo de Usuario  5. pasar a menu principal |
| Flujos alternos | 2.1. En caso de ingresar rut no valido, mostrar mensaje e ir a a 1  3.1 Si los datos no son ingresados, mostrar mensaje e ir al paso 1 |
| Pre-condiciones | 1. Cliente debe estar registrado como Usuario |
| Post-condiciones | Pasar al proceso de carga de menu. |

* En este caso tenemos que tomar en consideración que es clave que el usuario ingrese los datos de forma correcta al sistema, de lo contrario, en caso de que el rut ingresado sea incorrecta se volverá al punto 1.
* En el caso de que otro tipo de dato sea ingresado de forma incorrecta también se volverá al punto número 1.
* Si esto se repite se volverá sucesivamente al punto número 1 hasta que el usuario ingrese los datos correctamente.
* El Usuario que ingrese al Sistema debe estar registrado como Usuario por un Jefe de Zona o Administrador General, de lo contrario no podrá ingresa al Sistema.

|  |  |
| --- | --- |
| Nombre | Cargar menu |
| Autor | Sebastian Vargas |
| Actor | Usuario |
| Descripción | Proceso por el cual el sistema identifica y muestra el menu dependiendo, que rango tiene el usuario que interactua con el sistema . |
| Disparador | Login correcto |
| Flujo básico | 1. El sistema recibe los datos del usuario(Nombre, sector, rut, rango) 2. El sistema guarda que rango tiene el usuario 3. El sistema compara el rango y activa botones dependiendo que rango tiene el usuario |
| Flujos alternos |  |
| Pre-condiciones | 1. El usuario debe haber pasado por la pantalla de login |
| Post-condiciones | Mostrar menu de usuario |

* El Sistema comienza a procesar que tipo de usuario está accediendo a las características y funcionalidades, a qué podrá acceder y cuáles serán sus límites.
* Como requisito el Usuario debe haber ingresado correctamente sus datos y además haber estado registrado en el sistema por alguna autoridad en particular.
* A posterior, el Usuario podrá acceder a realizar consultas o lo que sea que necesite hacer dentro del Sistema.

|  |  |
| --- | --- |
| Nombre | Crear un usuario |
| Autor | Sebastian Vargas |
| Actor | Jefe de zona, Administrador general |
| Descripción | Proceso por el cual se procede a crear un usuario dentro del sistema . |
| Disparador | Botono Usarios |
| Flujo básico | 1. El usuario ingresa un rut a el campo “Rut”, 2. El usuario apreta el boton comprobar. 3. El sistema comprueba si el campo contiene datos. 4. El sistema comprueba si el usuario no existe. 5. El sistema activa los campos para que usuario introduzca los datos. 6. El usuario ingresa todos los campos requeridos y presion el boton “crear”. 7. El sistema comprueba que esten los campos completos. 8. El sistema crea el nuevo usuario. |
| Flujos alternos | 3.1 El usuario no ingreso datos y el sistema manda un mensaje con el error y el regresa al punto1  4.1 El rut ya esta registrado y lanza un mensaje que le informa al usuario de esto. El sistema habilita al opcion de editar y eliminar . Se regresa al punto 1  7.1 El usuario no completa todos los datos y el sistema manda un mensaje informando lo ocurrido, de regreso al punto 6 |
| Pre-condiciones | 1. El usuario debe haber cargado el menu |
| Post-condiciones | Se crea un usuario a la base de datos |

* En este proceso un Jefe Zona o Administrador General puede crear un nuevo Usuario en el Sistema.
* El flujo básico de este proceso es que esta autoridad ingresa todos los datos en lo que confiere a un nuevo Usuario.
* Se comprueba que los datos ingresados estén correctamente ingresados.
* Se comprueba que el Usuario que se va crear no está registrado anteriormente en el Sistema.
* Si todo fluye correctamente se confirma la creación del nuevo Usuario.
* El Sistema crea e ingresa este nuevo Usuario en el Sistema.
* En caso de no ingresar los datos correctamente el Sistema notificará este error y regresará al punto número 1.
* Si el Usuario que se está creando ya está en el Sistema, se comprobará este hecho y además se habilitarán 2 botones, “Editar”, “Eliminar”, estos funcionarán dependiendo de lo que determine el Usuario.
* Como requisito el Usuario debe haber realizado alguna consulta dentro del menú del Sistema.
* A posterior se crea un nuevo Usuario en la Base de Datos.

|  |  |
| --- | --- |
| Nombre | Eliminar un usuario |
| Autor | Sebastian Vargas |
| Actor | Jefe de zona, Administrador general |
| Descripción | Proceso por el cual se procede a eliminar un usuario dentro del sistema . |
| Disparador | Botono Usarios |
| Flujo básico | 1. El usuario ingresa un rut a el campo “Rut”, 2. El usuario apreta el boton comprobar. 3. El sistema comprueba si el campo contiene datos. 4. El sistema comprueba si el usuario no existe. 5. El sistema activa los campos para que usuario edite o elimine el usuario. 6. Usuario presiona boton “Eliminar” 7. El sistema elimina el usuario. |
| Flujos alternos | 3.1 El usuario no ingreso datos y el sistema manda un mensaje con el error y el regresa al punto1  4.1 El rut no esta registrado y lanza un mensaje que le informa al usuario de esto. El sistema habilita al opcion crear usuario . Se regresa al punto 1 |
| Pre-condiciones | 1. El usuario debe haber cargado el menu |
| Post-condiciones | Se elimina un usuario a la base de datos |

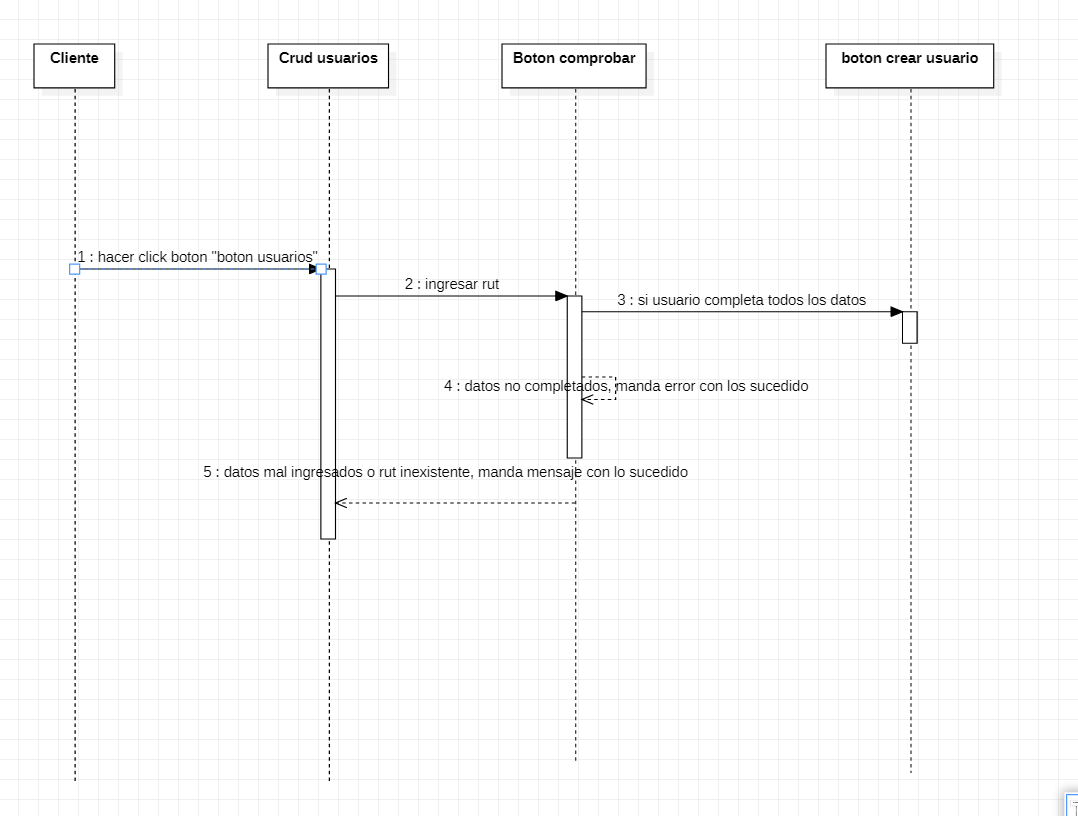
* Este proceso solo puede ser accionado por un tipo de Usuario “Jefe de Zona”, ó “Administrador General.”
* En este proceso el flujo corresponde a ingresa los datos de un Usuario que se desee eliminar del Sistema, con todos sus datos pertinentes.
* El Sistema va a corroborar que los campos contengan datos.
* El Sistema va a corroborar que el Usuario exista.
* El Sistema activará botones para la edición y eliminación del Usuario.
* En caso de que no se ingresen correctamente los datos se volverá al punto número1.
* En caso de que el Usuario no esté registrado se notifica esta situación al usuario y a posterior se activa un botón para la creación del Usuario.
* Como requisito el Usuario debe haber realizado alguna consulta dentro del menú del Sistema.
* A posterior se elimina Usuario en la Base de Datos.

|  |  |
| --- | --- |
| Nombre | Ingresar datos de delito |
| Autor | Sebastian Vargas |
| Actor | Jefe de zona, Administrador general, Operador |
| Descripción | Proceso por el cual se procede a ingresar ingreso de un delito dentro del sistema . |
| Disparador | Botono delitos |
| Flujo básico | 1. El usuario ingresa un codigo a el campo “codigo”. 2. El usuario apreta el boton comprobar. 3. El sistema comprueba si el campo contiene datos. 4. El sistema comprueba si el delito existe. 5. El sistema activa los campos para que usuario introduzca los datos. 6. El usuario ingresa todos los campos requeridos y presion el boton “crear”. 7. El sistema comprueba que esten los campos completos. 8. El sistema crea ingresa el nuevo delito. |
| Flujos alternos | 3.1 El usuario no ingreso datos y el sistema manda un mensaje con el error y el regresa al punto1  4.1 El delito esta registrado y lanza un mensaje que le informa al usuario de esto. El sistema habilita al opcion editar o elimnar delito . Se regresa al punto 1  7.1 El usuario le faltan datos por ingresar y el sistema manda un mensaje de lo sucedido. De vuelta al paso 5 |
| Pre-condiciones | 1. El usuario debe haber cargado el menu |
| Post-condiciones | Se elimina un usuario a la base de datos |

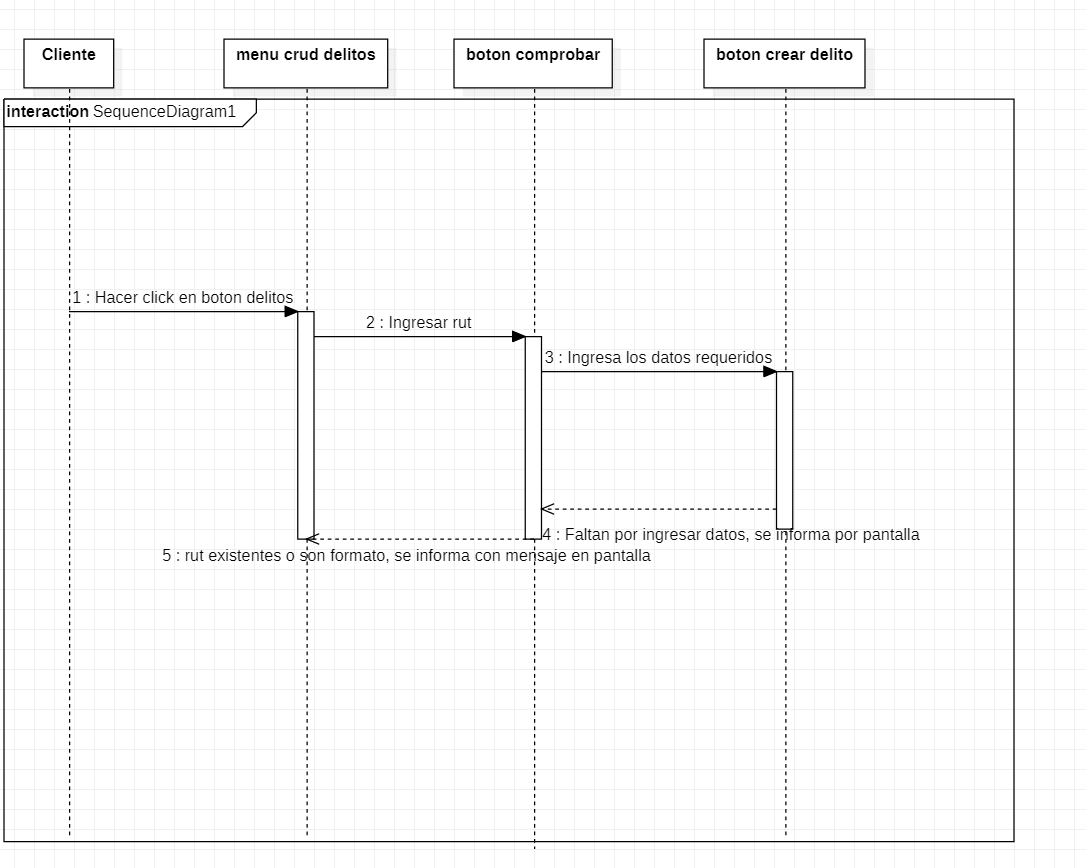
* Este proceso solo puede ser accionado por cualquier tipo de Usuario en el Sistema.
* Se ingresa primeramente el código del Delito.
* En este proceso el flujo corresponde a ingresa los datos de un Delito que se desee agregar Sistema, con todos sus datos pertinentes.
* El Sistema va a corroborar que los campos contengan datos.
* El Sistema va a corroborar que el Delito exista.
* En caso de que no se ingresen correctamente los datos se volverá al punto número1.
* En caso de que el Usuario no esté registrado se notifica esta situación al usuario y a posterior se activa un botón para la creación del Usuario.
* Como requisito el Usuario debe haber realizado alguna consulta dentro del menú del Sistema.
* A posterior se agrega un Delito en la Base de Datos.

*Diagramas de Secuencia detallados.*

* Ingreso al Sistema.



* Un Cliente ve el SplashScreen, en esta secuencia el cliente no realiza ninguna interacción.
* A continuación, se verifica que no haya errores en la base de datos, por consiguiente, aparece el menú de inicio de sesión donde se debe ingresar los datos requeridos, rut y contraseña.
* En caso de que los datos se hayan ingresado de forma correcta, se mostrará la pantalla de carga del menú y todas sus opciones dependiendo del tipo de usuario.
* En caso de que los datos ingresados no sean correctos o que no estén que falta algún dato, se muestra un mensaje indicando el error y volverá a la página de inicio de sesión para ingresar los datos nuevamente.
* Ingreso de Información de Delitos al Sistema.



* Se hace clic en el botón de delitos.
* Se ingresa el rut.
* Se ingresan los datos necesarios.
* En caso de que los datos ingresados sean incorrectos, se notifica al Usuario con un mensaje.
* En caso de que el rut no exista o el formato sea incorrecto, se notifica la Usuario con un mensaje.

*Modelo de Datos Detallado.*

Imagen que contiene texto, mapa

Descripción generada automáticamente

* *Sector*: ente fundamental para el desarrollo del Sistema. Todo va ligado a esta tabla. Por ejemplo: Un operador puede pertenecer a una institución y una institución debe pertenecer a un sector.
* Un sector puede tener varias instituciones, se unen por el código del sector.
* *Comuna*: en un sector hay varias comunas se unen a través del código del sector.
* Un sector puede tener varias comunas.
* *Delincuente*: tiene una comuna de residencia y un último lugar visto, ambas tablas diferentes entre sí.
* Cuando se comete un delito o se le realiza un control al delincuente, el último lugar visto de éste cambia a ser la comuna donde se registró un delito o se le hizo un control correspondiente.
* *Parentesco:* Un Delincuente puede tener parentesco con una gran cantidad de Delincuentes. En cambio, un Parentesco solo puede unir a dos Delincuentes.
* *Controles:* Una comuna puede tener varios controles, como también, un Delincuente puede tener varios controles en varias comunas.

*Interfaz de Programación de Aplicaciones.*

*PlaceHolder:*

* Texto Predictivo en determinados campos.

*MYSQL – JDBC:*

* Conexión con Base de Datos.

*Commons-Codec 1.14:*

* Encripta contraseñas en la página de inicio.

*JXL:*

* Genera documentos Excel para reportes.

*JCalendar:*

* Permite generar calendarios.

*JFreeChart:*

* Permite crear gráficos.

*IText 7:*

* Permite generar archivos PDF para Reportes.

*Conclusiones y Reflexiones.*

Consideramos que nuestra disposición hubiese sido aún mejor si un cliente real nos hubiese solicitado algún tipo de servicio, creemos esto bajo el fundamento de que quizá con un cliente real la comunicación con respecto a las necesidades y pensamientos de este serían de gran ayuda para nosotros trabajar. A pesar de todo el equipo está contento con los resultados obtenidos ya que los objetivos se lograron.

En una primera instancia nuestro Sistema estuve pensado de una forma muy diferente a la que resultó finalmente. El equipo tomó nota de los requisitos con mayor profundidad hasta reflexionar, que esta forma de desarrollar el Sistema era errónea. Por ello es por lo que se solventó con este nuevo planteamiento, que resolvió los objetivos de forma correcta.

Fue muy importante pensar en las limitaciones y funcionalidades a las que debía acceder cada tipo de Usuario, esto condicionó la gran mayoría del desarrollo del proyecto.

El equipo de trabajo tuvo algunas limitaciones al comienzo del proyecto, tales como falta de tiempo. Uno de los miembros estaba realizando su práctica laboral, esto limitó al equipo en algunas decisiones. En un principio nos planteamos realizar el proyecto en Flutter, se quiso realizar una aplicación móvil, un complemento a nuestro a proyecto que finalmente no se realizó por alcances de tiempo. Es por esta razón que se tomaron decisiones que se tomaron para el desarrollo del proyecto, decisiones que influyeron directamente en aspectos como: metodología, plazos entre tareas, tareas para cada integrante del grupo de trabajo.

Cabe mencionar que, al ser 2 miembros en el equipo las tareas que se designaron no siempre se realizaron por su encargado designado, en ocasiones ambos miembros trabajamos en tareas que estaban desarrolladas para un solo miembro.

A pesar de lograr los objetivos que nosotros nos planteamos, y a pesar de que logramos realizar las tareas y los procesos clave del proyecto como también solventar la problemática y lograr el desarrollo del Sistema. Como equipo consideramos que podríamos lograr resultados óptimos aún. El grupo de trabaja también considera e hecho de que el Sistema que presentaremos es bastante sencillo pero muy eficiente, no presenta errores de gran magnitud y presenta la fluidez necesaria para trabajar, además que es sencillo de comprender en su totalidad.

Nuestra solución con respecto a nuestra problemática puede ayudar de manera directa a las autoridades correspondientes. Nuestro Sistema puede ser de gran ayuda a la hora de prevenir crímenes, las herramientas que ofrecemos funcionan de una manera simple y están bien delimitadas para los distintos tipos de cargos que existen en las autoridades.

*Resultados Visuales del proyecto.*

*Administrador General.*

*Captura de pantalla de un celular

Descripción generada automáticamente*

*Jefe de Zona.*

*Captura de pantalla de un celular

Descripción generada automáticamente*

*Operador.*

*Captura de pantalla de un celular

Descripción generada automáticamente*

*Bibliografía.*

*API.*

* Commons-Codec-1.14. Última versión disponible desde Diciembre de 2019.

Sitio Web: <https://commons.apache.org/proper/commons-codec/>

* IText 7 (Versión de licencia gratuita). Escrito por Bruno Lowagie y Paulo Soares. Última versión disponible desde Diciembre de 2017.

Sitio Web: https://itextpdf.com/es

* JCalendar.

Sitio Web: <https://toedter.com/jcalendar/>

* JFreeCharts. Distribuida por David Gilbert. Última versión disponible desde Noviembre de 2017.

Sitio Web: <http://www.jfree.org/jfreechart/>

* JXL. Desarrollado por Andy Khan, Eric H. Jung.

Sitio Web: <http://jexcelapi.sourceforge.net/>

*Otros.*

* [*https://stackoverflow.com/questions/150646/how-do-i-create-a-new-excel-file-using-jxl*](https://stackoverflow.com/questions/150646/how-do-i-create-a-new-excel-file-using-jxl)
* [*https://es.stackoverflow.com/questions/44711/c%C3%B3mo-hacer-un-select-con-m%C3%BAltiples-condiciones*](https://es.stackoverflow.com/questions/44711/c%C3%B3mo-hacer-un-select-con-m%C3%BAltiples-condiciones)

*Anexos.*

*Código Fuente.*

public void crearReporteExcel(JTable tabla) throws IOException, WriteException {

// crea archivo excel como nombre la fecha actual

String ruta = ElegirRutaExcel();

File excel = new File(ruta);

WorkbookSettings conf = new WorkbookSettings();

conf.setEncoding("ISO-8859-1");

WritableWorkbook woorBook = Workbook.createWorkbook(new File(ruta), conf);

WritableSheet hoja1 = woorBook.createSheet("PREVCRIM", 0);

//cabezera

WritableFont titleFont = new WritableFont(WritableFont.ARIAL, 16, WritableFont.BOLD, true);

WritableCellFormat titleformat = new WritableCellFormat(titleFont);

WritableCellFormat titleformat1 = new WritableCellFormat(titleFont);

titleformat.setBackground(Colour.YELLOW);

titleformat.setBorder(Border.ALL, BorderLineStyle.MEDIUM);

titleformat1.setBorder(Border.ALL, BorderLineStyle.MEDIUM);

//hoja 1

//estadisticas

hoja1.addCell(new jxl.write.Label(0, 2, "PREVCRIM", titleformat1));

//tabla

DefaultTableModel tm = (DefaultTableModel) tabla.getModel();

for (int i = 0; i < tabla.getColumnCount(); i++) {

hoja1.addCell(new jxl.write.Label(i, 10, tabla.getColumnName(i)));

}

int valor1 = 0, valor2 = 0;

for (int i = 11; i < tabla.getRowCount() + 11; i++) {

for (int j = 0; j < tabla.getColumnCount(); j++) {

System.out.println("i: " + i + " J:" + j + " valor1: " + valor1 + " valor2: " + valor2);

hoja1.addCell(new jxl.write.Label(j, i, String.valueOf(tm.getValueAt(valor1, valor2))));

valor2++;

}

valor1++;

valor2 = 0;

}

woorBook.setProtected(true);

woorBook.write();

woorBook.close();

}

…

public void crearReportePDF(JTable tabla) throws IOException, DocumentException {

Document documento = new Document();

FileOutputStream ficheroPdf = new FileOutputStream(ElegirRutaPDF());

PdfWriter.getInstance(documento, ficheroPdf).setInitialLeading(1);

documento.open();

try {

Image foto = Image.getInstance("src/img/PREVCRIMLOGO.png");

foto.scaleToFit(100, 100);

foto.setAlignment(Chunk.ALIGN\_RIGHT);

documento.add(foto);

} catch (Exception e) {

e.printStackTrace();

}

documento.add(new Paragraph("Archivo generador automaticamente por sistema PREVCRIM",

FontFactory.getFont("arial", // fuente

14, // tamaño

Font.ITALIC, // estilo

BaseColor.BLACK))); // color

documento.add(new Paragraph(" ",

FontFactory.getFont("arial", // fuente

20, // tamaño

Font.ITALIC, // estilo

BaseColor.BLACK)));

PdfPTable tablaPDF = new PdfPTable(tabla.getColumnCount());

DefaultTableModel tm = (DefaultTableModel) tabla.getModel();

for (int i = 0; i < tabla.getColumnCount(); i++) {

tablaPDF.addCell(tabla.getColumnName(i));

}

for (int i = 0; i < tabla.getRowCount(); i++) {

for (int j = 0; j < tabla.getColumnCount(); j++) {

tablaPDF.addCell(String.valueOf(tm.getValueAt(i, j)));

}

}

documento.add(tablaPDF);

documento.close();

}

…

//en el caso de un administrador general

if (operador.getRango().equals("AdministradorGeneral")) {

if (this.ControlesDelincuente.getText().isEmpty() || tp.TryRut(this.ControlesDelincuente.getText())) {

// comprueba si el rut esta correcto

JOptionPane.showMessageDialog(null, "Ingrese rut valido");

} else if (!c.comprobarDelincuente(this.ControlesDelincuente.getText())) {

// comprueba si el delincuente está registrado en la base de datos

JOptionPane.showMessageDialog(null, "Delincuente no encontrado en la base de datos");

} else if (this.ControlesComuna.getText().isEmpty() || !tp.TryInt(this.DelincuentesComuna.getText())) {

// comprueba si la comuna está en el formato correcto

JOptionPane.showMessageDialog(null, "Ingrese una comuna valida");

} else if (!c.comprobarComuna(this.ControlesComuna.getText())) {

// comprueba si la comuna existe en la base de datos

JOptionPane.showMessageDialog(null, "No existe comuna en base de datos");

} else if (this.ControlesDireccion.getText().isEmpty()) {

// comprueba si la direccion contiene datos

JOptionPane.showMessageDialog(null, "Ingrese direccion");

} else if (this.ControlesFecha.getDate().toGMTString().isEmpty()) {

// comprueba si la fecha tiene datos

JOptionPane.showMessageDialog(null, "Ingrese fecha");

} else if (this.ControlesMotivo.getText().isEmpty()) {

// compruba si el motivo tiene datos

JOptionPane.showMessageDialog(null, "Ingrese motivo");

} else {

// si está todo correcto crea un control nuevo

String[] test = this.ControlesFecha.getDate().toGMTString().split(" ");

String dia = test[2] + "-" + getMonth(test[1]) + "-" + test[0];

c.crearControl(this.ControlesCodigo.getText(), this.ControlesDelincuente.getText(), this.ControlesComuna.getText(), this.ControlesDireccion.getText(), this.ControlesMotivo.getText(), dia);

// y actualiza ultimo lugar visto

c.crearControl2(this.ControlesDelincuente.getText(), this.ControlesComuna.getText());

}

} else // en el caso de un jefe de zona

if (operador.getRango().equals("JefeDeZona")) {

if (this.ControlesDelincuente.getText().isEmpty() || tp.TryRut(this.ControlesDelincuente.getText())) {

// comprueba si el rut esta correcto

JOptionPane.showMessageDialog(null, "Ingrese rut valido");

} else if (!c.comprobarDelincuente(this.ControlesDelincuente.getText())) {

// comprueba si el delincuente está registrado en la base de datos

JOptionPane.showMessageDialog(null, "Delincuente no encontrado en la base de datos");

} else if (this.ControlesComuna.getText().isEmpty() || !tp.TryInt(this.DelincuentesComuna.getText())) {

// comprueba si la comuna está en el formato correcto

JOptionPane.showMessageDialog(null, "Ingrese una comuna valida");

} else if (!c.comprobarComuna(this.ControlesComuna.getText(), operador.getRango())) {

// comprueba si la comuna existe en la base de datos y en especifico la zona del jefe de zonas

JOptionPane.showMessageDialog(null, "No existe comuna en base de datos o esa comuna no pertenece a su zona");

} else if (this.ControlesDireccion.getText().isEmpty()) {

// comprueba si la direccion contiene datos

JOptionPane.showMessageDialog(null, "Ingrese direccion");

} else if (this.ControlesFecha.getDate().toGMTString().isEmpty()) {

// comprueba si la fecha tiene datos

JOptionPane.showMessageDialog(null, "Ingrese fecha");

} else if (this.ControlesMotivo.getText().isEmpty()) {

// compruba si el motivo tiene datos

JOptionPane.showMessageDialog(null, "Ingrese motivo");

} else {

// si está todo correcto crea un control nuevo

String[] test = this.ControlesFecha.getDate().toGMTString().split(" ");

String dia = test[2] + "-" + getMonth(test[1]) + "-" + test[0];

c.crearControl(this.ControlesCodigo.getText(), this.ControlesDelincuente.getText(), this.ControlesComuna.getText(), this.ControlesDireccion.getText(), this.ControlesMotivo.getText(), dia);

// y actualiza ultimo lugar visto

c.crearControl2(this.ControlesDelincuente.getText(), this.ControlesComuna.getText());

}

} else // en el caso de un operador

{

if (this.ControlesDelincuente.getText().isEmpty() || tp.TryRut(this.ControlesDelincuente.getText())) {

// comprueba si el rut esta correcto

JOptionPane.showMessageDialog(null, "Ingrese rut valido");

} else if (!c.comprobarDelincuente(this.ControlesDelincuente.getText())) {

// comprueba si el delincuente está registrado en la base de datos

JOptionPane.showMessageDialog(null, "Delincuente no encontrado en la base de datos");

} else if (this.ControlesComuna.getText().isEmpty() || !tp.TryInt(this.DelincuentesComuna.getText())) {

// comprueba si la comuna está en el formato correcto

JOptionPane.showMessageDialog(null, "Ingrese una comuna valida");

} else if (!c.comprobarComuna(this.ControlesComuna.getText(), operador.getRango())) {

// comprueba si la comuna existe en la base de datos y en especifico la zona del operador

JOptionPane.showMessageDialog(null, "No existe comuna en base de datos o esa comuna no pertenece a su zona");

} else if (this.ControlesDireccion.getText().isEmpty()) {

// comprueba si la direccion contiene datos

JOptionPane.showMessageDialog(null, "Ingrese direccion");

} else if (this.ControlesFecha.getDate().toGMTString().isEmpty()) {

// comprueba si la fecha tiene datos

JOptionPane.showMessageDialog(null, "Ingrese fecha");

} else if (this.ControlesMotivo.getText().isEmpty()) {

// compruba si el motivo tiene datos

JOptionPane.showMessageDialog(null, "Ingrese motivo");

} else {

// si está todo correcto crea un control nuevo

String[] test = this.ControlesFecha.getDate().toGMTString().split(" ");

String dia = test[2] + "-" + getMonth(test[1]) + "-" + test[0];

c.crearControl(this.ControlesCodigo.getText(), this.ControlesDelincuente.getText(), this.ControlesComuna.getText(), this.ControlesDireccion.getText(), this.ControlesMotivo.getText(), dia);

// y actualiza ultimo lugar visto

c.crearControl2(this.ControlesDelincuente.getText(), this.ControlesComuna.getText());

}

}

…

// comprueba delito siendo administrador general

if (operador.getRango().equals("AdministradorGeneral")) {

// comprueba si el campo codigo tiene datos

if (!this.ControlesCodigo.getText().isEmpty() && tp.TryInt(this.ControlesCodigo.getText())) {

// comprueba si el control existe

if (c.comprobarControl(this.ControlesCodigo.getText())) {

// si existe muestra mensaje en pantalla

JOptionPane.showMessageDialog(null, "Control existente, proceda a eliminar o modificar");

// habilita campos a rellenar

this.ControlesCodigo.setEditable(false);

this.controlesActivar();

// habilita el boton editar y eliminar

this.ControlesCrear.setEnabled(false);

this.ControlesEditar.setEnabled(true);

this.ControlesEliminar.setEnabled(true);

} else {

// sin no existe muestra mensaje en pantalla

JOptionPane.showMessageDialog(null, "Control no existente, proceda a crear uno");

// habilita campos a rellenar

this.controlesActivar();

// habilita el boton crear

this.ControlesEditar.setEnabled(false);

this.ControlesEliminar.setEnabled(false);

this.ControlesCrear.setEnabled(true);

}

} else {

JOptionPane.showMessageDialog(null, "Ingrese un codigo numerico");

}

} else // en el caso que sea un jefe de zona

if (operador.getRango().equals("JefeDeZona")) {

// comprueba que el campo no esta vacio y es numerico

if (!this.ControlesCodigo.getText().isEmpty() && tp.TryInt(this.ControlesCodigo.getText())) {

// comprueba si el control es de alguna de las comunas pertenecientes al sector

if (c.comprobarControl(ControlesCodigo.getText(), operador.getZona())) {

// si pertenece al sector muestra mensaje en pantalla y permite eliminar o editar

JOptionPane.showMessageDialog(null, "Control existente, proceda a eliminar o modificar");

// habilita campos a rellenar

this.ControlesCodigo.setEditable(false);

this.controlesActivar();

// habilita el boton editar y eliminar

this.ControlesCrear.setEnabled(false);

this.ControlesEditar.setEnabled(true);

this.ControlesEliminar.setEnabled(true);

} else {

// si no existe en la zona, busca en todos las zonas

if (c.comprobarControl(ControlesCodigo.getText())) {

// si existe ya en otra zona muestra en pantalla

JOptionPane.showMessageDialog(null, "No tienes permisos para ver este control");

} else {

// si no existe en ninguna zona, da camino a crear el control// sin no existe muestra mensaje en pantalla

JOptionPane.showMessageDialog(null, "Control no existente, proceda a crear uno");

// habilita campos a rellenar

this.controlesActivar();

// habilita el boton crear

this.ControlesEditar.setEnabled(false);

this.ControlesEliminar.setEnabled(false);

this.ControlesCrear.setEnabled(true);

}

}

} else {

JOptionPane.showMessageDialog(null, "Ingrese un codigo numerico");

}

} else {

// la ultima opcion es operador, solo puede crear control

if (!this.ControlesCodigo.getText().isEmpty() && tp.TryInt(this.ControlesCodigo.getText())) {

// comprueba si el control existe

if (c.comprobarControl(this.ControlesCodigo.getText())) {

// si existe muestra mensaje en pantalla

JOptionPane.showMessageDialog(null, "No tienes permisos para ver este control");

} else {

// sin no existe muestra mensaje en pantalla

JOptionPane.showMessageDialog(null, "Control no existente, proceda a crear uno");

// habilita campos a rellenar

this.controlesActivar();

// habilita el boton crear

this.ControlesEditar.setEnabled(false);

this.ControlesEliminar.setEnabled(false);

this.ControlesCrear.setEnabled(true);

}

} else {

JOptionPane.showMessageDialog(null, "Ingrese un codigo numerico");

}

}