**Desarrollo para Sistemas Distribuidos**

**Equipo:** DSD 201300 C61A JAVATEAM

**Repositorio:** <https://github.com/jromucho/JAVATEAM>

**Scrumy :** http://www.scrumy.com/JAVATEAM

**Integrantes:**

* José Carlos Romucho Sotelo - U800093
* Victor Sotelo Elias - U201100023
* Pedro Alberto Balvin Lozano - U201021196
* Percy Ameri Maita - U201014064
* Daniel Peña Cueva - U201200939

Horario: 07:30 - 10:30 PM

Profesor: José Díaz Díaz

Ciclo: VI

**Índice**

[1. Introducción 3](#_Toc346829215)

[2. Antecedentes 3](#_Toc346829216)

[3. Objetivo del proyecto 4](#_Toc346829217)

[4. Beneficios del proyecto 4](#_Toc346829218)

[*a) Servicios débilmente acoplados: 4*](#_Toc346829219)

[*b) Abstracción de servicios: 4*](#_Toc346829220)

[*c) Reusabilidad 4*](#_Toc346829221)

[*d) Homologación 4*](#_Toc346829222)

[5. Descripción del proceso 4](#_Toc346829223)

[*a) Actividades 4*](#_Toc346829224)

[*b) Reglas del negocio 5*](#_Toc346829225)

[*c) Historia de Usuario 5*](#_Toc346829226)

[*I. Login de usuarios (Lenguaje PHP) 5*](#_Toc346829227)

[*II. Mostrar ofertas del día (Metodología WS - SOA) 5*](#_Toc346829228)

[*III. Registrar Usuario (Lenguaje PHP) 5*](#_Toc346829229)

[*IV. Actualizar Datos Usuario (Lenguaje PHP) 6*](#_Toc346829230)

[*V. Realizar Compra (Metodología WS - SOA) 6*](#_Toc346829231)

[*VI. Consulta de Compras (Metodología WS - REST) 6*](#_Toc346829232)

[*VII. Listado de Empresas (Metodología SOA) 6*](#_Toc346829233)

[6. Arquitectura del sistema 6](#_Toc346829234)

[7. Desarrollo del sistema 10](#_Toc346829235)

[*a) Presentación de la Aplicación : 10*](#_Toc346829236)

[*b) Inicio de Sesión: 10*](#_Toc346829237)

[*c) Registrar Usuario : 11*](#_Toc346829238)

[*d) Registro de usuario Exitoso: 11*](#_Toc346829239)

[*e) Registrar Reclamo : 12*](#_Toc346829240)

[*f) Listar Reclamo : 12*](#_Toc346829241)

[8. Conclusiones 13](#_Toc346829242)

# Introducción

En el presente documento se detalla el trabajo elaborado en base a las conocimientos y metodologías adquiridos en el curso de Desarrollo de Sistemas Distribuidos. En el presente trabajo se elaborado un sistema de Libro de Reclamación que permitirá al usuario ingresar y consultar sus quejas. Se ha implementado un Servicio Web con la Arquitectura Orientada a Servicios (SOA), donde el usuario podrá realizar consultas sobre sus quejas y estas internamente consultan los datos de 03 entidades del estado (Reniec, Ministerio de Justicia y Poder Judicial)

# Antecedentes

En el Perú de acuerdo al Decreto Supremo N° 011-2011-PCM, en el caso de los proveedores que desarrollen actividades económicas de servicios públicos regulados o que se encuentren bajo la supervisión de la Superintendencia de Banca, Seguros y AFP, el procedimiento previamente establecido por el organismo supervisor para la atención de quejas y reclamos de los consumidores deberá entenderse como la implementación y puesta a disposición del Libro de Reclamaciones, siempre y cuando este procedimiento permita dejar constancia de la presentación del reclamo o la queja, de su contenido y que además regule el cómputo de los plazos de atención, así como 3la puesta a disposición de canales para su presentación. En estos casos los proveedores se regirán por las normas emitidas por dichos organismos.

En el caso de las empresas supervisadas por la Superintendencia de Banca, Seguros y AFP, los Sistemas de registros de quejas y reclamos deberán encontrarse a disposición inmediata y accesible al consumidor y permitir que éstos obtengan una copia o constancia de la queja o reclamo presentado, con la información mínima prevista del presente Reglamento. Asimismo, deberán exhibir en un lugar visible, como mínimo un Aviso del Libro de Reclamaciones en cada una de sus agencias, utilizando el formato establecido del presente Reglamento. Por ende, toda proveedor deberá contar con un *Libro de Reclamación (*documentofísico o virtual) en la cual los consumidores podrán registrar quejas o reclamos sobre los productos o servicios ofrecidos en un determinado establecimiento comercial.

Por tal motivo, el presente trabajo se elaborará un Sistemas de Libro de Reclamación la cual permitirá al Ministerio de Justicia , cumplir con el Decreto Supremo N° 011-2011-PCM para lo cual pondrá a disposición de los usuarios este Sistema que permitirá registrar sus reclamos, quejas y/o sugerencias.

Para el Sistema de Libro de Reclamación se tendrá que validar que los datos de los usuarios sean correctos (se validará dicha información con la Reniec ) así como que dicho usuario no cuente con antecedente penales (se validará con la Policía Nacional del Perú) y judiciales (Poder Judicial).

Para la cual, la necesidad comunicar e intercambiar información con las demás identidades (Reniec, Poder Judicial, Policía Nacional del Perú), origina la necesidad de implementar un Sistema Distribuido la cual utilizará la implementación de Web Services SOAP como medio de comunicación entre dichas aplicaciones .

# Objetivo del proyecto

El objetivo del presente trabajo es poder implementar un Sistema de Libro Reclamaciones , para poder brindarle al consumidor una herramienta eficiente para poder presentar su reclamo o queja en la oportunidad y lugar que ocurren los hechos, lo cual a su vez otorgará al Ministerio de Justicia la posibilidad de solucionar el reclamo o queja de manera inmediata (en línea), sin necesidad de la intervención de la autoridad administrativa.

# Beneficios del proyecto

1. Servicios débilmente acoplados:

En el proyecto, los servicios web expuestos son independientes del cliente que los está llamando, es decir la aplicación en PHP, ya que los resultados que mostramos pueden ser entendidos por cualquier otra tecnología.

1. Abstracción de servicios:

Nuestro proyecto maneja clases e interfaces las cuales ayudan a mantener aislada la lógica del negocio y los procesos de los servicios en sí, es decir exponemos interfaces, pero la lógica está en las clases.

1. Reusabilidad

Como nuestro proyecto utiliza servicios web, como basados en SOA estos pueden reutilizarse fácilmente y pueden convertirse en nuevas aplicaciones compuestas.

1. Homologación

Al reutilizar un servicio se homologan funciones que son soportadas por el mismo servicio, proporcionando ya sea una misma lógica de negocio o interfaz.

# Descripción del proceso

* + 1. **Actividades**

Como Usuario:

1. Ingresar a la pagina del ministerio de justicia / libro de reclamaciones

2. Registrarse como usuario

3. Iniciar Sesión

4. Registrar los reclamos, quejas o sugerencias.

5. Consultar en su bandeja de entrada las respuestas a sus reclamos, quejas Y/o sugerencias.

* + 1. **Reglas del negocio**

1. Para poder registrar un reclamo, queja o sugerencia, el usuario debe registrarse en la aplicación ingresando sus datos personales.
2. El usuario podrá ver las respuestas de sus reclamos, quejas y/o sugerencias en su bandeja de entrada (después de iniciar sesión), los reclamos serán respondidos en un periodo máximo de 30 días a partir del registro del mismo.
3. Al responder un reclamo, el ministerio envía un correo al usuario que su reclamo ha sido atendido (correo que registro al crear la cuenta) y esta a su vez podrá ser impreso desde el libro de reclamación virtual del Ministerio de Justicia.
4. El cliente puede actualizar sus datos y/o contraseña de acceso, validando la cuenta de correo electrónico proporcionado en el registro inicial.

* + 1. **Historia de Usuario**

1. **Registrar Usuario (Lenguaje Java - Metodología WS / SOA)**

Como usuario deseo registrar mis datos para poder hacer utilizar el Libro de Reclamación Virtual del MINJUS publicada en la web de la entidad.

Para realizar esta consulta tenemos:

* Si el DNI no existe mostrará un mensaje de error.
* Si los datos se encuentran incompletos, mostrará un mensaje de error.
* Si el email ya existe, mostrará un mensaje de error.
* Si el formato de email es incorrecto, mostrará un mensaje de error.
* Si los datos son correctos, mostrará un mensaje de confirmación.

1. **Login de usuarios (Lenguaje Java)**

Como usuario deseo loguearme en el libro de reclamación virtual del Ministerio de Justicia publicada en la web de la entidad, para poder tener acceso de registrar un reclamo , queja o sugerencia, Para ello, se dispone lo siguiente:

* Cuando ingreso mal el usuario, se mostrará un mensaje de error “El usuario no existe”.
* Cuando ingreso mal la contraseña, se mostrará un mensaje de error “La contraseña no es correcta , intenta nuevamente por favor”.
* Cuando el usuario se encuentra bloqueado, se mostrará un mensaje de error “Su cuenta está bloqueado debido a los 03 intentos fallidos de la contraseña, intente en 30 minutos nuevamente por favor”.
* Cuando sea el usuario y contraseña correcta, se mostrará un mensaje de confirmación “Ud. Ha ingresado sus credenciales correctamente”.

1. **Actualizar Datos Usuario (Lenguaje Java)**

Como usuario deseo actualizar mis datos y/o credenciales, para así mantener información real en todo momento. Para realizar esta consulta tenemos:

* Si los datos se encuentran incompletos, mostrará un mensaje de error.
* Si el email ya existe, mostrará un mensaje de error.
* Si el formato de email es incorrecto, mostrará un mensaje de error.
* Si los datos son correctos, mostrará un mensaje de confirmación.

1. **Mostrar estado de los reclamos, quejas y/o sugerencia (Metodología WS - SOA)**

Como usuario deseo revisar el estado de las reclamaciones que he realizado .Para realizar esta consulta tenemos:

* Ingresar fecha, la cual devolverá todas las reclamaciones que se encuentren dentro del rango de fechas especificado.
* Ingresar fecha, si no existen reclamos realizados que se encuentren dentro del rango de fechas, esta no devolverá ninguna información.
* Ingresar el número del reclamo realizado, si no existe el reclamo no devolverá ninguna información.

1. **Registrar reclamo, queja y/o sugerencia (Metodología WS - SOA)**

Como usuario deseo registrar un reclamo, queja y/o sugerencia en el libro de reclamación virtual de MINJUS, de acuerdo a la atención y/o servicio que este considere que no ha sido el más adecuado. Para realizar esta consulta tenemos:

* + - * Ingresar los datos de la atención recibida, si falta datos, mostrará un mensaje de error.
      * Seleccionar el tipo de reclamo, si no se selecciona dicho dato mostrará un mensaje de error.
      * Ingresar descripción del reclamo, si falta datos, mostrará un mensaje de error.
      * Si la fecha de atención está dentro de rango de validez, devolverá un mensaje de confirmación.

# Arquitectura del sistema

Como condiciones de partida, se plantea que la arquitectura debe considerar que la aplicación debe:

* Ser modular y distribuida.
* Una parte de la arquitectura debe estar orientada a Servicios, es decir, debe ser un aplicativo para aplicativos, no orientada a usuario final.
* Debe presentar tanto un interfaz (o conjunto de interfaces) como un api (o conjunto de apis) que faciliten tanto la comunicación como la creación de nuevos interfaces.
* Como parte de la arquitectura debe existir una web que consuma los servicios creados.

Posteriormente, consideramos la información a manejar en:

* Los datos y/o registros que conforman la base de la información manipulable, en este caso las ofertas de cada empresa.
* Las *reglas* que contienen la definición de la lógica de negocio.
* El *flujo de trabajo* (workflow) que incorpora el despliegue de actividades a realizar en base al tiempo.

Entonces, desde un punto de vista funcional, se plantea un arquitectura modular basada en nodos. Los cuales definimos que son: Servidor, la Base de datos y los distintos Clientes que consumen los servicios, para el presente trabajo del Libro de Reclamaciones Virtual se esta considerando el lenguaje Java con base de datos PostgreSQL

**Servicios implementados**

En el presente trabajo se tuvo la necesidad de implementar un servicio web para consultar y/o validar los datos de los usuarios en los registros de la RENIEC, el web services está basado en SOAP.

También en se implemento un servicio web basado en SOAP para consultar si el usuario que se está registrando tiene antecedentes penales, la aplicación se conecta vía SOAP a los aplicativos del Poder Judicial.

**Tecnologías implementadas**

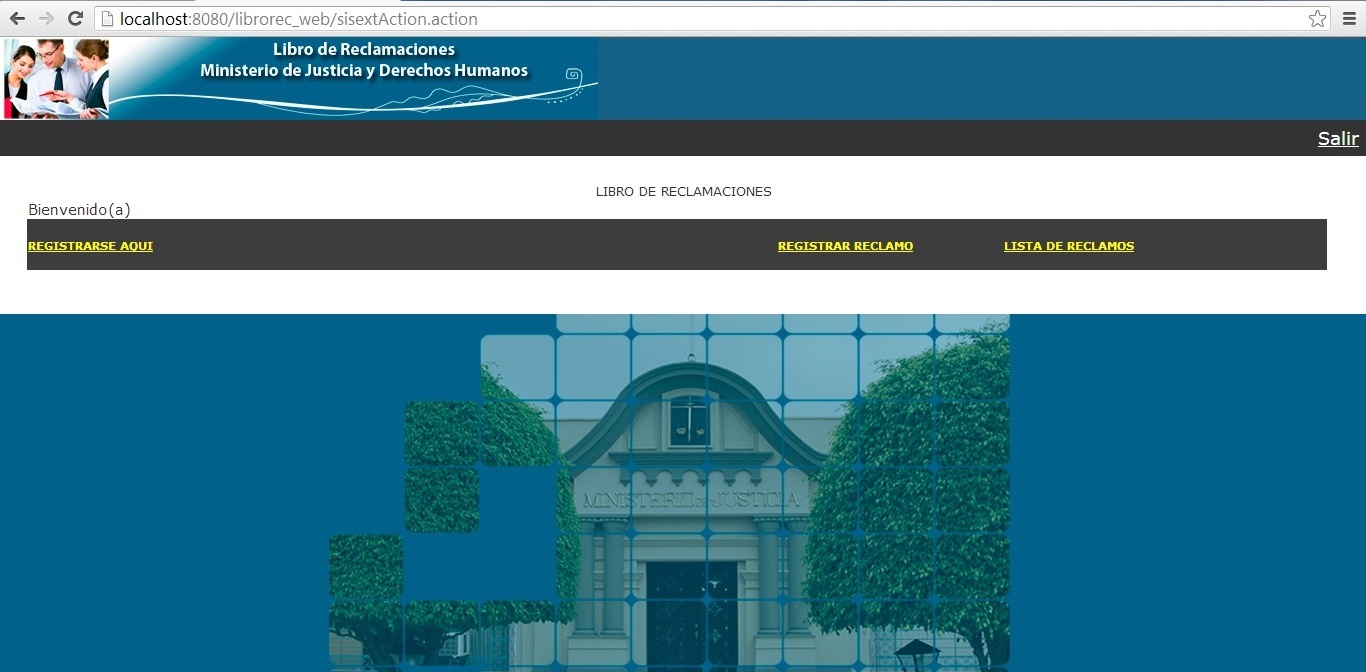
* PostgreSQL
* Java
* Junit 4.8
* Visual Estudio 2010
* Sql Server 2008
* IDE: Spring tool Suite 9

**DIAGRAMA DE DESPLIEGUE**

****

# Desarrollo del sistema

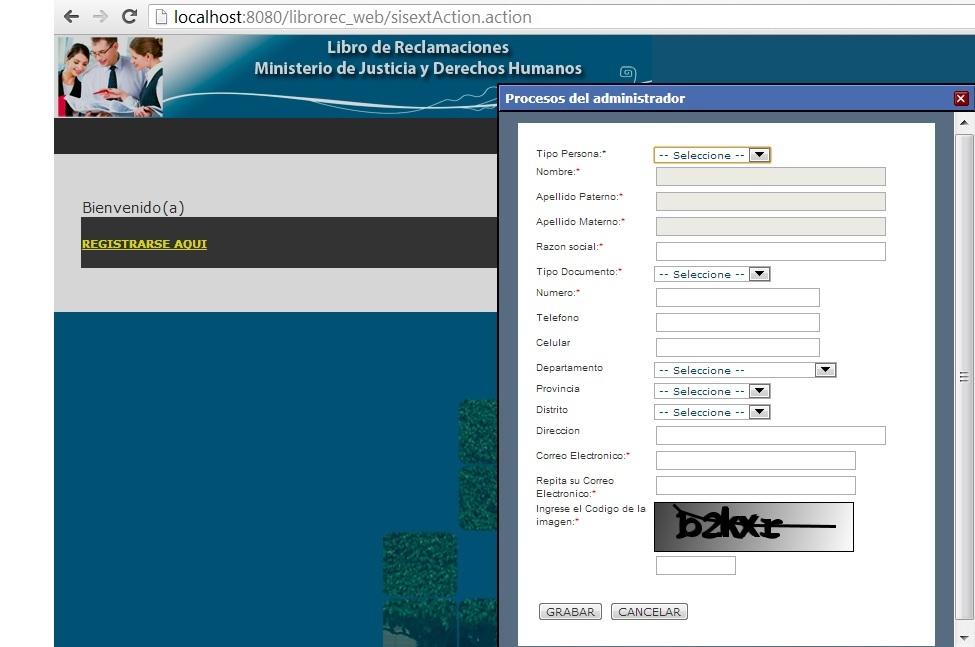
1. Presentación de la Aplicación :



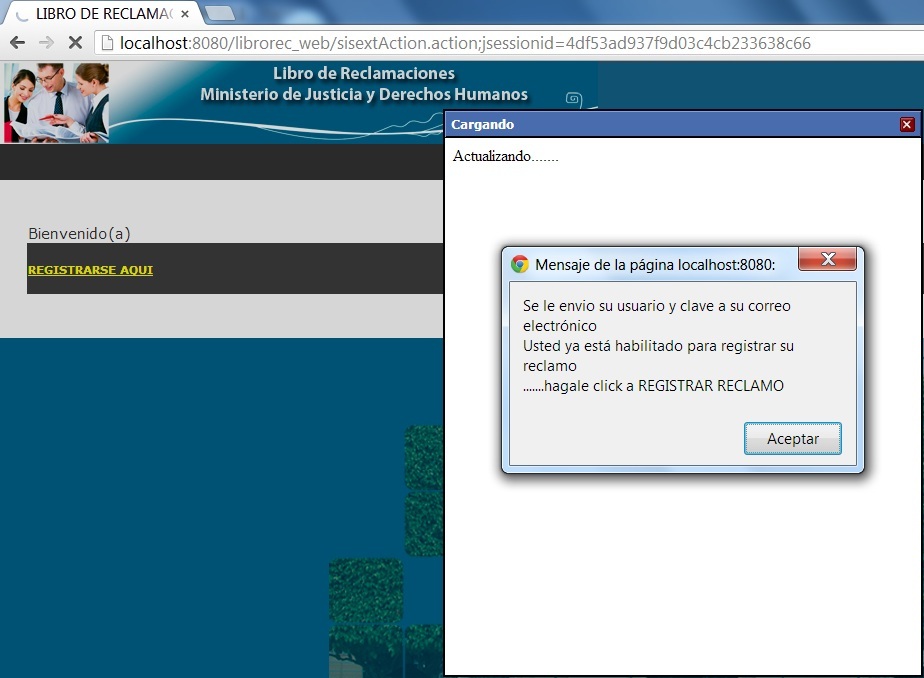
1. Inicio de Sesión:



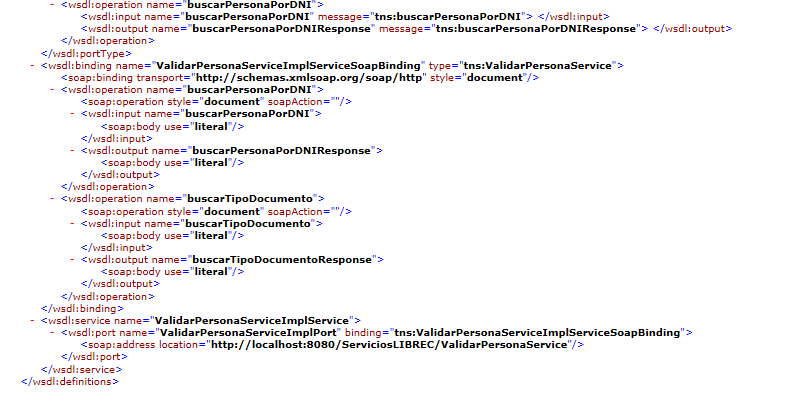
1. Registrar Usuario :



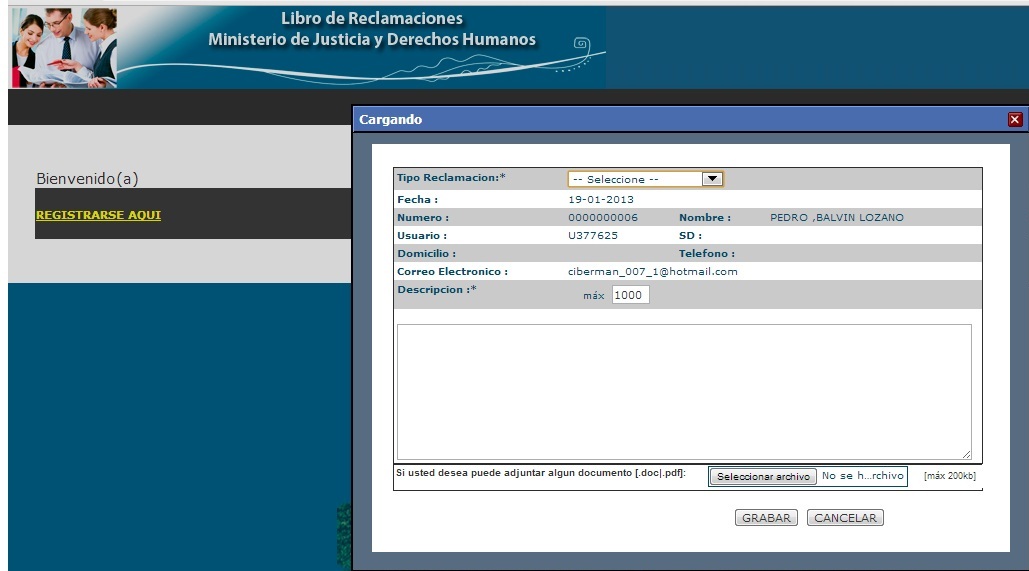
1. Registro de usuario Exitoso:



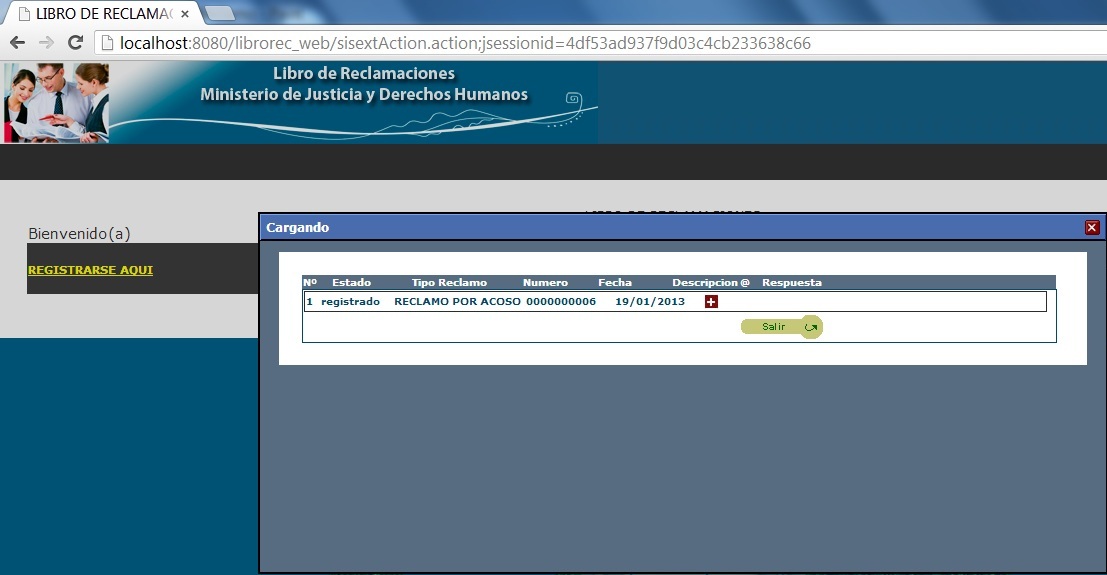
1. XML Validar Persona:



1. Registrar Reclamo :



1. Listar Reclamo :



# Conclusiones

1.- La facilidad de implementar Servicios web podemos hacer que las aplicaciones que consumen dichos servicios sean independientes de la tecnología y plataforma en las cuales e ha implementado.

2.- Github nos ayudó a trabajar en equipo de manera ordenada y versionada nuestro proyecto, utilizando repositorios locales en donde se realizaban cambios, para luego subirlos al repositorio central y confirmar dichos cambios del proyecto.

3.- Scrumy nos ayuda a identificar las iteraciones por historia de usuarios y ver el avance y progreso de los mismos, lo que nos permitió poder desarrollar el presente trabajo en forma ordenada y proporcionada.

4.- Se logró implementar el presente trabajo un Libro de Reclamación Virtual basado en la necesidad del MINJUS, para que los usuarios hagan sus reclamos. Los servicios web creados se conectan a una base de datos SQL 2008 mediante SOAP (Reniec).

**CRITERIOS A EVALUAR POR EL PROFESOR**

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Puntaje** |  |
| C1: El proyecto está creado, tiene la arquitectura definida y se encuentra versionado en el repositorio de código. | 2 |  |
| C2: El proyecto tiene desarrollada la capa de presentación completa utilizando el patrón MVC para al menos una User Story. | 2 |  |

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Puntaje** |  |
| C3: El equipo demuestra trabajo colaborativo, todos los miembros realizan commit sobre el repositorio de código semanalmente, se comunican efectivamente, logran auto-organizarse y completan la retrospectiva. | 2 |  |
| C4: El documento tiene todas las secciones completas, aplica los conceptos vistos en clase, sustenta adecuadamente con correcta ortografía y gramática. | 2 |  |
| C5: El proyecto tiene desarrollada la capa de presentación completa utilizando el patrón MVC para todas las User Stories. | 4 |  |
| C6: El proyecto tiene desarrollado al menos una User Story completa utilizando Web Services SOAP. Cuenta con tests de integración para probar el servicio. | 4 |  |
| C7: El equipo realiza una presentación de 10 minutos mostrando el resultado del Sprint con exposición fluida de ideas. La presentación inicia puntualmente, todos los miembros participan activamente y responden las preguntas. | 4 |  |