

ESCUELA DE INGENIERÍA EN SISTEMAS

Tema:

Desarrollo de una aplicación web para el seguimiento del proceso de declaraciones tributarias en empresas de Asesoría Contable y Tributaria empleando el modelo MVC5.

Disertación de grado previo a la obtención del título de

Ingeniero de Sistemas y Computación

Línea de investigación:

Ingeniería de Software y/o Plataformas Educativas.

Autor:

LUIS FRANCISCO TORRES DIER

Director:

DARÍO JAVIER ROBAYO JÁCOME, ING. MSC.

Ambato-Ecuador

Octubre 2016

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE AMBATO

HOJA DE APROBACIÓN

Tema:

Desarrollo de una aplicación web para el seguimiento del proceso de declaraciones tributarias en empresas de Asesoría Contable y Tributaria empleando el modelo MVC5.

Línea de Investigación:

Ingeniería de Software y/o Plataformas Educativas.

Autor:

LUIS FRANCISCO TORRES DIER

Darío Javier Robayo Jácome, Ing. Msc. f._____
CALIFICADOR

José Marcelo Balseca Manzano, Ing. Msc. f._____
CALIFICADOR

Santiago Alejandro Acurio Maldonado, Ing. Msc. f._____
CALIFICADOR

Teresa Freire Aillón, Ing. Msc. f._____
DIRECTORA DE UNIDAD ACADÉMICA

Hugo Rogelio Altamirano Villaroel, Dr. f._____
SECRETARIO GENERAL PUCESA

Ambato-Ecuador

Octubre 2016

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Luis Francisco Torres Dier portador de la cédula de ciudadanía No. 1803019189-9 declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniero de Sistemas y Computación son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.

Luis Francisco Torres Dier
CI. 180301918-9

AGRADECIMIENTO

Agradezco a mi madre quién ha sido el apoyo fundamental durante toda mi etapa de estudiante y toda mi vida sabiendo como encaminarme siempre por el camino del bien y con principios y valores fundamentales para estar hoy aquí, de la misma forma agradezco a mis maestros, que fueron los pilares que fundamentaron mi educación no sólo académica, sino también personal y profesional predicando con el ejemplo y fomentado el espíritu de investigación y amor con respeto a la profesión, en especial al Ing. Msc. Darío Robayo Jácome director de este trabajo de disertación por su paciencia y sabios consejos a lo largo del desarrollo de la misma y durante toda la carrera y por último pero siempre lo más importante le agradezco a Dios por darme la voluntad y fuerza de lucha para culminar esta etapa de mi vida de manera satisfactoria.

DEDICATORIA

Esta tesis va dedicada principalmente a mi madre por ser el apoyo incondicional en mi vida y siempre buscar los mayores beneficios para mí, brindándome todo lo que estuvo al alcance de sus manos para convertirme en una persona de bien, de la misma manera se la dedicó a mi tío Marco Antonio Dier Paredes por ser siempre más que un tío, sino un amigo, un padre y un gran ejemplo a seguir de una persona que no se rinde jamás ante las adversidades, a su vez quiero dedicarle esta tesis a mi madre quien ha sido la persona que siempre ha estado en los momentos que más la eh necesitado siendo un pilar fundamental en mi vida y brindándome siempre ese apoyo extra que necesité en todo momento de dificultad y en general a todas las personas que hicieron que está meta se pudiera cumplir.

RESUMEN

Este proyecto de investigación fue desarrollado con el fin de crear un sistema de control que le permita a una empresa de Asesoría Tributaria y Contable, llevar un seguimiento más preciso de las declaraciones realizadas por cada contribuyente perteneciente a la misma. La aplicación fue implementada mediante el modelo MVC5 el cual permite un desarrollo dirigido a cumplir con los requerimientos. Se crearon dos módulos para el manejo del sistema, el primero para la administración de todos los datos necesarios y la información requerida en la elaboración de las declaraciones así como para su ingreso y seguimiento; y el segundo para que los contribuyentes pudieran conocer el estado actual de cada una de sus declaraciones, tomando en cuenta las diferentes obligaciones que estos tienen para con el SRI. Finalmente se implementó un sistema de envío automático de correos electrónicos, que tendrán como finalidad informar al cliente mediante una notificación el momento en el que se registra dentro del sistema una declaraciones, a su vez se incorporó la opción de que los contribuyentes puedan descargar un resumen completo de todas las declaraciones que se hayan registrado hasta la fecha.

Palabras Clave: aplicación, MVC5, asesoría tributaria, declaraciones, contribuyentes.

ABSTRACT

This research project was developed in order to create a follow-up system that allows an accounting and tax consultancy to have previse monitoring of the tax statements of every tax-payer that belong to the company. The application was implemented using a MVC5 model towards the fulfillment of the requirements. Two modules were created to handle the system, the first one to manage all required data and the information that is needed to prepare tax statements as well as for data entry and follow up; the second one is for the tax-payers to get information about the current status of the statements, taking into consideration their obligations with the IRS. Finally an automated mailing system was implemented in order to keep the client posted by a notification once the statement is registered in the application. Tax-payers can also download a complete summary of the statements that they have registered to date.

Key words: *application, MVC5, tax consultancy, tax statements, tax-payer.*

ÍNDICE DE CONTENIDOS

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD iii
AGRADECIMIENTO iv
DEDICATORIA v
RESUMEN vi
ABSTRACT vii
CAPÍTULO I. 2
MARCO TEORICO 2
1.1. Antecedentes 2
1.2. Problema 3
1.2.1. Descripción del problema 3
1.2.2. Preguntas básicas 4
1.3. Justificación 4
1.4. Objetivos 4
1.4.1. Objetivo General 4
1.4.2. Objetivos Específicos 4
1.5. Fundamentos Teóricos 5
1.5.1. HTML 5 5
1.5.1.1. ¿Qué es HTML 5? 5
1.5.1.2. Algoritmo de Perfilado de HTML5 6
1.5.1.3. Cabeceras en HTML5 6
1.5.1.4. Seccionado Implícito 7
1.5.1.5. Web Socket 8
1.5.1.6. Recursos sin Conexión 8
1.5.1.7. API de Almacenamiento Web 9
1.5.1.8. Interfaces del Almacenamiento Web 10
1.5.1.9. Base de Datos Indexada 10
1.5.2. ASP.NET MVC 5 11
1.5.2.1. ¿Qué es ASP.Net MVC 5? 11
1.5.2.2. Ventajas de ASP.Net MVC5 11
1.5.2.3. Identidad de ASP.NET 11

1.5.2.4. Plantillas de bootstrap 12
1.5.2.5. Ruteo de los Atributos 14
1.5.2.6. Andamiaje de ASP.Net 14
1.5.2.7. Filtros de autenticación 15
1.5.2.8. ASP.Net MVC y sus convenciones 15
1.5.2.9. Convención sobre Configuración 16
CAPITULO II. 17
METODOLOGIA 17
2.1. Métodos 17
2.1.1. Método Analítico 17
2.1.2. Método Experimental 17
2.1.3. Entrevista 18
2.1.3.1. Resultados de las entrevistas 18
2.2. Metodología para el desarrollo de la aplicación 19
2.2.1. Análisis de Requisitos 20
2.2.2. Diseño Conceptual 20
2.2.3. Diseño Navegacional 21
2.2.4. Diseño de Presentación 21
3.1. Análisis de Requerimientos 22
3.1.1. Entrevista para análisis de requerimientos 22
3.1.2. Documento de Requerimientos 26
3.1.2.1. Introducción 26
3.1.2.1.1. Propósito 26
3.1.2.1.2. Visión General 26
3.1.2.1.3. Descripción General 26
3.1.2.2. Requerimientos Específicos 26
3.1.2.2.1. Requerimientos Funcionales 26
3.1.3. Diagramas Casos de Uso 28
3.1.3.1. Contribuyente Obligado Llevar Contabilidad 28
3.1.3.2. Generalización Declaración 28
3.1.3.3. Generalización de Documentos 29
3.1.3.4. Generalización Elaboración de Declaraciones 29
3.1.3.5. Resumen Contribuyente Obligado a Llevar Contabilidad 29

3.1.3.6. Contribuyente No Obligado a Llevar Contabilidad 30
3.1.3.7. Generalización Declaración 30
3.1.3.8. Generalización Entrega de Documentos..... 30
3.1.3.9. Elaboración Declaraciones 31
3.1.3.10. Resumen..... 31
3.1.3.11. IVA por Pagar Fuente 32
3.1.3.12. Resumen..... 32
3.1.4. Diagramas Casos de Uso del Sistema 32
3.1.4.1. Registro de Cliente 32
3.1.4.2. Administración Fechas..... 34
3.1.3.3. Registro Declaraciones..... 35
3.1.3.4. Envío de Mail 36
3.2. Fase II Diseño Conceptual..... 38
3.3. Modelo de Datos (Diseño de Datos) 38
3.3.1. Diccionario de Datos..... 39
3.3.2. Tablas Específicas..... 39
3.4. Diseño de Navegación..... 42
3.4.1. Diagrama de Componentes 42
3.4.2. Modelo de Navegación 42
3.4.2.1. Simbología Utilizada 42
3.4.2.2. Nuevo Cliente 43
3.4.2.3. Cliente Existente..... 44
3.4.2.4. Administración 44
3.4.2.5. Declaraciones 45
3.4.2.6. Tipos de Declaración 45
3.4.2.7. Usuarios..... 46
3.4.2.8. Clientes..... 46
3.5. Modelo de Presentación 47
3.5.1. Diagrama de Presentación..... 47
3.5.1.1. Registro de un Cliente..... 47
3.5.1.2. Registro de un Usuario 48
3.5.1.3. Registro de Tipo de Declaración..... 48
3.1.5.4. Registro de Declaración..... 49

CAPÍTULO IV 50
RESULTADOS 50
4.1. Aplicación 50
4.1.1. Login 50
4.1.2. Página Principal Perfil Administrador 51
4.1.3. Página Principal Perfil Usuario 51
4.1.4. Registro Usuarios 52
4.1.5. Listado Usuarios 52
4.1.6. Edición Usuarios 53
4.1.7. Registro de Cliente 53
4.1.8. Listado de Clientes 54
4.1.9. Edición de Cliente 54
4.1.10. Ingreso Tipo de Declaración 55
4.1.11. Listado Tipo de Declaración 55
4.1.12. Edición Tipo de Declaración 56
4.1.13. Ingreso Declaración 56
4.1.14. Listado de Declaraciones 57
4.1.15. Exportar Reporte de Declaraciones 57
CAPITULO V 58
CONCLUSIONES Y RECOMENDACIONES 58
5.1. Conclusiones 58
5.2. Recomendaciones 58
REFERENCIAS 60
ANEXOS 61

INDICE DE ILUSTRACIONES

Ilustración 1.1. Algoritmo de Perfilado de HTML5 [5] 6
Ilustración 1.2. Cabeceras en HTML5 [5] 7
Ilustración 1.3. Seccionado Implícito 7
Ilustración 1.4. Web Socket [5] 8
Ilustración 1.5. Proyecto MVC base [6] 13
Ilustración 1.6. Proyecto MVC5 base [6] 13
Ilustración 3.1. Contribuyente Obligado a Llevar 28
Ilustración 3.2. Generalización Declaración 28
Ilustración 3.3. Generalización de Documentos 29
Ilustración 3.4. Generalización Elaboración de Declaraciones 29
Ilustración 3.5. Contribuyente no Obligado a llevar Contabilidad 30
Ilustración 3.6. Generalización Declaración 30
Ilustración 3.7. Generalización Entrega de Documentos 30
Ilustración 3.8. Elaboración Declaraciones 31
Ilustración 3.9. IVA por Pagar 32
Ilustración 3.10. Registro de Cliente 32
Ilustración 3.11. Registro Declaración 35
Ilustración 3.12. Envío de Mail 36
Ilustración 3.13. Diseño Conceptual 38
Ilustración 3.14.. Modelo de Datos 38
Ilustración 3.15. Diseño de Navegación 42
Ilustración 3.16. Diagrama UWE Navegación Sistema Principal 43
Ilustración 3.17. Diagrama UWE Nuevo Cliente 43
Ilustración 3.18. Diagrama UWE Paciente Existente 44
Ilustración 3.19. Diagrama UWE Administración 44
Ilustración 3.20. Diagrama UWE Declaraciones 45
Ilustración 3.21. Diagrama UWE Tipo de Declaración 45
Ilustración 3.22. Diagrama UWE Usuario 46
Ilustración 3.23. Diagrama UWE Usuario 46
Ilustración 3.24. Diagrama UWE Formulario Cliente 47
Ilustración 3.25. Diagrama UWE Formulario Usuario 48
Ilustración 3.26. Diagrama UWE Formulario Tipo de Declaración 48
Ilustración 3.27. Diagrama UWE Formulario Declaraciones 49
Ilustración 4.1. Login 50
Ilustración 4.2. Página Principal 51
Ilustración 4.3. Página Principal Perfil 51
Ilustración 4.4. Registro Clientes 52
Ilustración 4.5. Listado Usuarios 52
Ilustración 4.6. Edición Usuarios 53
Ilustración 4.7. Registro de Cliente 53

Ilustración 4.8. Listado de Clientes 54
Ilustración 4.9. Edición de Cliente 54
Ilustración 4.10. Tipo de Declaración 55
Ilustración 4.11. Listado Tipo de Declaración 55
Ilustración 4.12. Edición Tipo de Declaración 56
Ilustración 4.13. Registro de Declaración 56
Ilustración 4.14. Listado de Declaraciones 57
Ilustración 4.15. Reporte de Declaraciones 57

INTRODUCCIÓN

Dentro del ámbito contable la elaboración de las declaraciones tributarias representa un pilar fundamental en nuestra sociedad cada vez mejor administrada y controlada en aspectos legales, el mantenimiento de la información de los clientes, su disponibilidad y veracidad se tornan a priori el punto eje para conseguir un mejor manejo de este tipo de acciones. Las empresas tributarias y contables de la ciudad de Ambato y en forma más directa sus contadores públicos autorizados (CPA) requieren de una forma rápida, segura y sencilla de comunicarse con sus clientes para poder solicitarles todos los documentos e información necesarios para cada tipo de declaración que se necesite realizar, de la misma forma los clientes requieren de una herramienta o un medio que les permita estar informados sobre el estado en el que se encuentran sus procesos contables actuales, por ello se presenta la necesidad de mantener un registro o un control de seguimiento de cada cliente y de las declaraciones que se lleven a cabo en un período de tiempo determinado, buscando siempre la factibilidad de que sea autónomo y únicamente requiera que se le alimente para actualizar los datos que se han receptado.

CAPÍTULO I.

MARCO TEORICO

1.1. Antecedentes

Las soluciones en sistemas computacionales realizados de acuerdo a los requerimientos específicos de las empresas, brindan a las mismas un mejor nivel de producción y desempeño, los avances tecnológicos que se van desarrollando traen grandes ventajas en todos los ámbitos donde se implementan, ayudando a automatizar tareas y aligerar el tiempo de trabajo del hombre. [1]

Basado en el desarrollo de aplicaciones para web empleando el modelo MVC como es el caso de [2] se ha determinado que el principal propósito es reducir el esfuerzo de programación y su principal característica es se puede tomar al Modelo, sus vistas y controladores como partes diferentes e independientes de un todo que a su vez colaboran entre sí, pues todo cambio que se realice o se produzca en la parte del modelo se verá reflejado de manera inmediata en las vistas correspondientes de la aplicación.

La innovación alcanzada en [3] al elaborar la maquetación de todo un sitio web en el lenguaje HTML5 demuestra que la incorporación de este lenguaje dentro del desarrollo de aplicaciones web presenta un claro avance tanto en el ámbito estético como funcional de la interfaz con la que se relaciona el cliente para con la misma.

1.2. Problema

1.2.1. Descripción del problema

La empresa de Asesoría Tributaria MD tiene como sus principales problemas el control de entrada y salida de información requerida para elaborar las declaraciones como son los ingresos y gastos obtenidos por la empresa en el período correspondiente, así como retenciones y facturas de cada cliente en caso de ser requeridas, además de la falta de control de fechas para la declaración específica de cada uno de ellos, debido a la falta de un sistema que automatice el control de estos ámbitos el servicio que se brinda a los clientes puede llegar a tener fallos puesto que la falta de comunicación entre la empresa y sus clientes en cuanto al estado actual de las declaraciones puede ocasionar que no se pueda llevar a cabo las mismas en los periodos de tiempo correspondientes, lo que trae consigo que no se tenga una administración adecuada.

Actualmente los negocios no se encuentran adaptados hacia la automatización de procesos y los avances tecnológicos que se van desarrollando, por lo que se presenta una amplia posibilidad de implementación de nuevos sistemas que permitan y ayuden al mejoramiento en el rendimiento tanto de tiempo como de actividades realizadas por parte de los empleados.

El presente proyecto tiene como finalidad la automatización del seguimiento del proceso de declaración tributaria dentro de la empresa de Asesoría Tributaria MD, brindando un sistema fiable y confiable que permita evitar retraso en las declaraciones de los clientes así como permitir a los mismos

conocer el estado actual de las mismas, además permitirá que la empresa se contacte previamente con el cliente solicitando la información necesaria para elaborar dicha declaración.

1.2.2. Preguntas básicas

- ¿Cómo aparece el problema que se pretende solucionar? No Aplica.
- ¿Por qué se origina? No aplica.
- ¿Qué lo origina? El control y manejo manual de la información para la gestión de los clientes y seguimiento en el proceso de declaraciones dentro de la empresa de Asesoría Tributaria MD.
- ¿Cuándo se origina? No aplica.
- ¿Dónde se origina? No Aplica.
- ¿Dónde se detecta? No Aplica.

1.3. Justificación

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar una aplicación web para el seguimiento en el proceso de declaraciones tributarias en empresas de Asesoría Contable y Tributaria empleando el modelo MVC5.

1.4.2. Objetivos Específicos

- Fundamentar teóricamente sobre el modelado MVC5.
- Diagnosticar la situación actual de la empresa de Asesoría Contable y Tributaría MD.

- Diseñar una aplicación web para el seguimiento en el proceso de declaraciones tributarias en base al modelo MVC5.

1.5. Fundamentos Teóricos

1.5.1. HTML 5

1.5.1.1. ¿Qué es HTML 5?

HTML 5 es la siguiente generación de HTML, reemplazando a sus antecesores HTML 4.01, XHTML 1.0, y XHTML1.1, HTML 5 provee nuevas características necesarias para el desarrollo de aplicaciones web modernas, además de estandarizar muchas de las características de la plataforma web que los desarrolladores han venido utilizando desde hace ya varios años.

Al igual que sus predecesores HTML 5 está desarrollado para ser multiplataforma sin importar si se emplea sistemas operativos Windows, Mac OS X o Linux se puede aprovechar de todas las ventajas que posee HTML 5, lo único indispensable para poder emplearlo es tener actualizado el navegador de internet.

Las últimas versiones de los navegadores más conocidos como son Safari, Google Chrome, Mozilla Firefox y Opera ya cuentan con soporte para HTML 5 en su gran mayoría, de la misma forma los navegadores que se encuentran en los teléfonos móviles poseen un gran nivel de aceptación para con el lenguaje de HTML 5 y sus diversas características. [4]

1.5.1.2. Algoritmo de Perfilado de HTML5

Dentro de HTML5 las secciones pueden ubicarse de manera anidada, todo contenido incluido dentro del elemento <body> es parte de una sección, los límites de cada sección son definidos explícita o implícitamente (Ilustración 1), las secciones definidas explícitamente son el contenido definido en las etiquetas <body>, <section>, <aside>, <footer>, <header> y <nav>. [5]

```
<section>
  <h1>Forest elephants</h1>
  <section>
    <h1>Introduction</h1>
    <p>In this section, we discuss the lesser known forest elephants.
  </section>
  <section>
    <h1>Habitat</h1>
    <p>Forest elephants do not live in trees but among them.
  </section>
  <aside>
    <p>advertising block
  </aside>
</section>
<footer>
  <p>(c) 2010 The Example company
</footer>
```

Ilustración 1.1. Algoritmo de Perfilado de HTML5 [5]

1.5.1.3. Cabeceras en HTML5

Dentro de cada documento todos los perfiles requieren de una cabecera para poder ser útiles dentro de los mismos, dichas cabeceras se basan en una regla básica la cual determina (Ilustración 2), “el primero elemento de cabecera HTML (<h1>, <h2>, <h3>, <h4>, <h5>, <h6>) define la cabecera de la sección actual”, estos elementos cuentan con un rango dado determinado por el nombre de cada elemento, donde <h1> tiene el rango más alto y <h6> el más bajo, la estructura de cada sección es la que determina el perfil, más no el rango de cabecera de las mismas. [5]

```
<section>
  <h1>Forest elephants</h1>
  <p>In this section, we discuss the lesser known forest elephants.
    ...this section continues...
  <section>
    <h2>Habitat</h2>
    <p>Forest elephants do not live in trees but among them.
        ...this subsection continues...
  </section>
</section>
<section>
  <h3>Mongolian gerbils</h3>
  <p>In this section, we discuss the famous mongolian gerbils.
     ...this section continues...
</section>
```

Ilustración 1.2. Cabeceras en HTML5 [5]

1.5.1.4. Seccionado Implícito

Los elementos de seccionado de HTML5 al no ser obligatorios para la definición de un perfil, para mantener la compatibilidad con la web dominada por HTML4, emplean el seccionado implícito (Ilustración 3), para definir secciones sin ellas. Los elementos de cabecera cuando no son el primer encabezado de sus secciones establecen un nuevo seccionado implícito y la forma en que esta sección implícita se posiciona dentro del perfil se define mediante su rango relativo con la cabecera que le precede, si es de rango más bajo abre una sub-sección implícita de la sección.

```
<section>
  <h1>Forest elephants</h1>
  <p>In this section, we discuss the lesser known forest elephants.
    ...this section continues...
  <h3 class="implicit subsection">Habitat</h3>
  <p>Forest elephants do not live in trees but among them.
    ...this subsection continues...
</section>
```

Ilustración 1.3. Seccionado Implícito

1.5.1.5. Web Socket

Al ser una tecnología experimental, no cuenta con una especificación estabilizada, por lo que se debe revisar una tabla de prefijos para determinar cuál es el adecuado para cada navegador, de la misma manera se debe establecer que la sintaxis y el comportamiento de esta tecnología pueden estar sujetos a cambios en las versiones futuras de cada navegador a la vez que estos van cambiando.

El objeto *"WebSocket"* (Ilustración 4), provee la API para crear y gestionar la conexión de WebSocket hacia el servidor, así como para enviar y recibir datos en la conexión. [5]

```
WebSocket WebSocket(
  in DOMString url,
  in optional DOMString protocols
);

WebSocket WebSocket(
  in DOMString url,
  in optional DOMString[] protocols
);
```

Ilustración 1.4. Web Socket [5]

1.5.1.6. Recursos sin Conexión

HTML5 provee un mecanismo de cache de aplicación que permite que las aplicaciones realizadas en base de web sigan funcionando aun cuando estén desconectados, los desarrolladores pueden emplear la cache de la aplicación para especificar los recursos que los navegadores deberán mantener en cache y permanecer disponibles para los usuarios que no se

encuentren conectados. Las aplicaciones que se han mantenido en cache cargan y funcionan correctamente incluso cuando los usuarios dan click en el botón de refrescar cuando se encuentren desconectados.

Usar el cache de la aplicación brinda los siguientes beneficios:

- Navegación fuera de línea: Los usuarios pueden navegar en el sitio incluso cuando se encuentran desconectados de internet.
- Velocidad: Los recursos almacenados en cache son locales por lo que cargan de manera más rápida.
- Carga reducida del servidor: Los navegadores únicamente descargan recursos que hayan cambiado desde el servidor. [5]

1.5.1.7. API de Almacenamiento Web

La API de almacenamiento web provee mecanismos con los cuales los navegadores pueden almacenar de manera más segura y mucho más intuitiva los datos clave que han sido empleados por las cookies.

Existen dos mecanismos incorporados dentro del almacenamiento web que son:

- sessionStorage: Mantiene un área de almacenamiento separada para cada origen dado que se encuentre disponible durante la sesión de la página, es decir mientras el navegador se encuentre abierto, incluyendo la recarga de páginas.
- localStorage: Realiza la misma función pero permanece cuando el navegador es cerrado y se vuelve a abrir.

1.5.1.8. Interfaces del Almacenamiento Web

- **Almacenamiento**

Le permite establecer, recuperar y eliminar datos de un dominio específico y de un tipo de almacenamiento (sesión o local).

- **Window**

La API de almacenamiento web extiende el objeto “window” con dos propiedades nuevas (window.sessionStorage y window.localStorage) las cuales proveen acceso a la sesión actual del dominio y a los objetos locales almacenados respectivamente.

- **StorageEvent**

El StorageEvent es disparado en los objetos “window” del documento cuando un área de almacenamiento cambia.

1.5.1.9. Base de Datos Indexada

La base de datos indexada es un sistema de base de datos transaccional, basada en Java-Script orientada a objetos, la cual permite almacenar y recuperar objetos que se encuentren indexados mediante una clave; cualquier objeto soportado por el algoritmo de clonación estructurado puede ser almacenado. Se requiere de la especificación del esquema de base de datos, abrir una conexión a la base de datos, actualizar y recuperar los datos mediante una serie de transacciones. [5]

1.5.2. ASP.NET MVC 5

1.5.2.1. ¿Qué es ASP.Net MVC 5?

ASP.Net es la plataforma de desarrollo de aplicaciones web presentada por Microsoft bajo el conocido patrón Modelo – Vista – Controlador, de acceso gratuito y que viene incluido dentro del paquete de Visual Studio y cuenta con un interesante grupo de nuevas características incluidas dentro de la colección de herramientas del programador web.

1.5.2.2. Ventajas de ASP.Net MVC5

- Facilidad para el mantenimiento de aplicaciones.
- Facilidad para realizar testeos unitarios automatizados y desarrollo orientado a pruebas.
- URL limpias y fáciles de recordar.
- Control absoluto sobre el HTML resultante generado.
- Posibilidad de crear web “responsive” usando plantillas framework Bootstrap de forma nativa.
- Gran integración con JQuery y otras bibliotecas JavaScript.
- Magnifico rendimiento y escalabilidad.
- Gran extensión y flexibilidad.

1.5.2.3. Identidad de ASP.NET

Los sistemas de membresía y autenticación en MVC5 han sido completamente reescritos como parte del nuevo sistema de identidad de ASP. NET, este nuevo sistema se mueve mucho más allá que unas simples restricciones desactualizadas del previo sistema de membresías de

ASP.NET, mientras que añade cierta sofisticación y configurabilidad al sistema simple de membresías que inicio con MVC4.

- Fue diseñado para funcionar a través de la familia de ASP.Net (MVC, Formularios Web, Páginas Web, API Web, SignalR y aplicaciones hibridas usando cualquier combinación de estas).
- Hace que el almacenamiento de la información adicional de los usuarios (números de cuenta, información de contacto) sea tan fácil como añadir propiedades a la clase modelo que representa al usuario.
- Al ser diseñada usando interfaces la API de identidad de ASP.Net permite escribir unidades de prueba para el código de la aplicación relacionado con el usuario.
- En lugar de únicamente concentrarse en la autenticación mediante usuario y contraseña la API de identidad de ASP.Net entiende que los usuarios comúnmente son autenticados mediante proveedores sociales como puedes ser cuentas de Microsoft, Facebook o Twitter. [6]

1.5.2.4. Plantillas de bootstrap

El diseño visual de la plantilla por defecto de MVC1 como se puede observar en la (Ilustración 5) se ha mantenido casi sin variar hasta MVC3, cuando se crea un nuevo proyecto de MVC y se lo ejecuta, se obtiene un cuadrado blanco en un fondo azul.

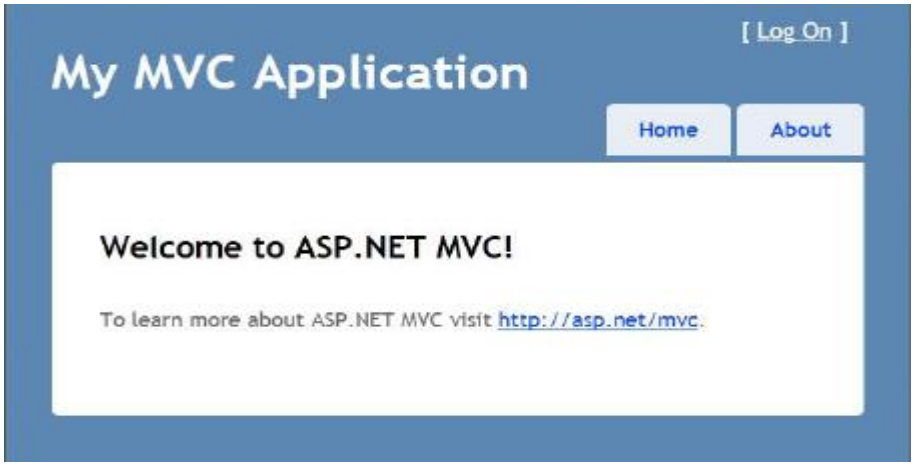


Ilustración 1.5. Proyecto MVC base [6]

En MVC5 las plantillas de los proyectos se movieron a ejecutarse en el popular marco de bootstrap (Ilustración 6), el cual fue creado por un desarrollador y diseñador en twitter por lo que el diseño actual para MVC5 actualmente se ve como algo que requiere ser desplegado antes de la producción. [6]

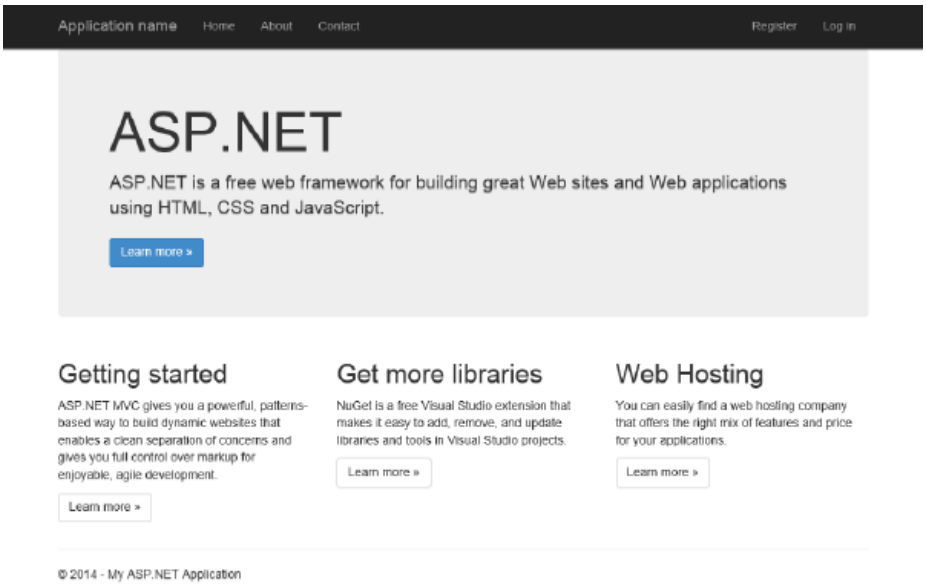


Ilustración 1.6. Proyecto MVC5 base [6]

1.5.2.5. Ruteo de los Atributos

El ruteo de los atributos es una nueva opción para la especificación de rutas mediante la colocación de anotaciones dentro de los controladores de clases o métodos de acciones.

El ruteo de atributos dentro del marco de MVC ASP. Net tiene dos propósitos principales:

- Coincidir con las solicitudes entrantes que de otra manera no coincidirían con ningún archivo dentro del sistema de archivos y mapear las solicitudes hacia un controlador de acción.
- Construir URL de salida que correspondan con los controladores de acción. [6]

1.5.2.6. Andamiaje de ASP.Net

Es el proceso de generar código repetitivo basado en las clases del modelo, MVC ha tenido este andamiaje desde su primera versión, pero era limitado a proyectos de MVC, el nuevo sistema de andamiaje de ASP.Net funciona con cualquier aplicación de ASP.Net, adicionalmente incluye un soporte para la construcción de poderosos andamiajes personalizados completos con diálogos personalizados y una API de andamiaje comprensiva.

De la misma manera puede generar el código repetitivo necesario para crear, leer, actualizar y eliminar funcionabilidad en una aplicación, las plantillas de andamiaje pueden examinar el tipo de definición para cada modelo y generar un controlador, vistas asociadas para el controlador y en algunos casos clases para el control de datos.

El proceso de andamiaje sabe que nombre dar a los controladores, a las vistas, que código debe ir dentro de cada componente y dónde ubicar estas piezas dentro del proyecto para que la aplicación funcione correctamente. [6]

1.5.2.7. Filtros de autenticación

MVC ha soportado desde hace ya mucho tiempo una característica denominada filtros de autorización, la que permite restringir el acceso a un controlador o acción basándose en los roles de membresía u otra lógica personalizada, sin embargo, hay una gran diferencia entre autenticación y autorización, por lo que los recientemente añadidos filtros de autenticación ejecutan antes los filtros de autorización, permitiendo al usuario el acceso que requiere y que la API de identidad de ASP.Net provee para ejecutar una lógica de autenticación personalizada y propia de cada usuario.

1.5.2.8. ASP.Net MVC y sus convenciones

Las aplicaciones realizadas en ASP.Net MVC, por defecto tienen un gran número de características propias o convenciones, esto permite a los desarrolladores el evitar tener que configurar y especificar información o configuraciones que pueden ser inferidas basadas en dichas convenciones.

MVC emplea una estructura de nombres de directorio basada en convenciones cuando se trata de resolver las vistas de las plantillas, y estas permiten omitir la el uso de referencias del camino de origen donde se encuentran dichas vistas.

MVC se encuentra diseñado alrededor de algunas convenciones base por defecto sensibles que pueden ser sobrescritas de ser necesario, este

concepto es comúnmente conocido como “convención sobre configuración”. [6]

1.5.2.9. Convención sobre Configuración

El concepto de convención sobre configuración hace algunos años atrás y de manera esencial significa: “*Nosotros sabemos, hasta ahora, como elaborar una aplicación web. Vamos a rotar esa experiencia en un marco para de esta forma no tener que configurar absolutamente todo de nuevo”.*

Se puede ver este concepto funcionando en ASP.Net MVC al mirar a las tres directrices principales que hicieron funcionar a la aplicación:

- Controladores
- Modelos
- Vistas

Las convenciones de ASP.Net MVC son bastante sencillas, lo que se espera dentro de la estructura de una aplicación MVC es lo siguiente:

- El nombre de cada clase del controlador termina con la palabra “*Controller*”.
- Existe un directorio simple de vistas para todas las vistas dentro de una aplicación.
- Las vistas que los controladores emplean se encuentran ubicadas dentro de un subdirectorio dentro del directorio principal de las vistas y son nombradas de acuerdo al nombre del controlador que las utiliza. [6]

CAPITULO II.

METODOLOGIA

2.1. Métodos

En el presente proyecto se utilizaron los siguientes métodos de investigación:

2.1.1. Método Analítico

Se denomina método analítico a aquel que se basa en la descomposición de las partes o elementos de un todo, con el objetivo de poder observar y analizar las causas, la naturaleza y los efectos del objeto que se estudia para llegar a comprender su esencia. [7]

Para este proyecto se empleó un proceso de investigación bibliográfica, la misma que cuenta con una amplia y meticulosa búsqueda de información, así como de análisis de la misma y selección de las partes que mayor nivel de aporte brindaron para el desarrollo de la aplicación.

2.1.2. Método Experimental

Se denomina método experimental a aquel método que se basa en la recopilación de datos para su posterior comparación, tomando en cuenta las variables que interactúan en el mismo se las puede denominar variables dependientes (las variables que se busca medir), variables independientes (variables que son manipuladas por el encargado del experimento) y por último las variables extrañas (otras variables que pueden influir). [8]

Por este método se lleva a cabo las pruebas respectivas a la aplicación como es la comprobación del envío de mails, la validación del proceso de seguimiento y los tiempos de respuesta de la aplicación, así como su estabilidad y efectividad.

2.1.3. Entrevista

Se denomina entrevista al acto de comunicación oral que se lleva a cabo entre 2 o más personas (entrevistado(s) y entrevistador), buscando obtener alguna información u opinión tomando en cuenta que los resultados de la misma no serán para el entrevistador sino para terceros que pendientes de la entrevista. [9]

Se realizará una entrevista a la gerente de la empresa de Asesoría Tributaria MD para conocer cómo se encuentran la información actual de los clientes, así como los requerimientos necesarios para el desarrollo de la aplicación.

2.1.3.1. Resultados de las entrevistas

Tras la entrevista realizada se tuvo como resultados información muy importante, la misma que como punto principal se obtuvo que la empresa actualmente realiza todos los procesos de manera manual, lo que brinda la oportunidad propicia para la implementación de este sistema para su uso y funcionamiento de forma inmediata.

2.2. Metodología para el desarrollo de la aplicación

En el presente proyecto se utilizará la metodología de desarrollo de Lenguaje Unificado de Modelado UML (Unified Modeling Language) en su modalidad web denominada Ingeniería Web basada en UML UWE (UML-based Web Engineering), puesto que brinda una forma detallada del cómo desarrollar una aplicación web exhaustiva en el proceso de diseño incluyendo flujos de trabajo y puntos de control. [10]

UWE es una metodología dirigida a la especificación de aplicaciones tomando como principal punto de apoyo en la personalización de sus características, tales como la creación de un modelo específico para los usuarios o una sección exclusiva para el establecimiento de las características necesarias en función de las preferencias, conocimiento o tareas del usuario. [10]

La principal razón para el uso de UWE en lugar de propias técnicas de modelado es la aceptación que posee UML dentro del desarrollo de software, la flexibilidad para la definición de un dominio web específico y un amplio apoyo en el modelado visual por parte de las herramientas de casos de UML ya existentes. [11]

UWE es una guía para elaborar y desarrollar aplicaciones web, también considerado como una extensión de UML ya que emplea mecanismos de este. Los modelos más exclusivos son: el modelo navegacional y el modelo de presentación, así como también emplea diagramas como casos de uso, modelo conceptual, diagrama de clases entre los más utilizados. [12]

Las fases que describen el uso de la metodología UWE son los siguientes:

2.2.1. Análisis de Requisitos

La etapa de análisis de requerimientos consiste de dos partes, la realización de los casos de uso de la aplicación y sus relaciones, y las actividades que describen en detalle a los casos de uso. [11]

Los casos de uso son aquellos que modelan la funcionalidad de una aplicación o sistema, mostrando como los actores externos pueden interactuar con dicha aplicación, teniendo como finalidad saber quiénes son los participantes y cuáles serán los módulos o casos de uso que entrarán en esta interacción. [12]

En esta etapa se procede a reunir de manera detallada las características de funcionalidad que debe tener la aplicación, diferenciando entre las necesidades de información, navegación, adaptación y las interfaces de usuario, basándose en el análisis de los casos de uso generados.

2.2.2. Diseño Conceptual

La etapa se desarrolla los diagramas de clases comunes de UML considerando los requisitos reflejados en los casos de uso. [11]

Los diagramas de clases se representan las clases del sistema o aplicación y las relaciones entre estas, este diagrama se genera tomando en cuenta las propiedades que posee cada clase en la programación orientada a objetos (abstracción, herencia, encapsulamiento y agregación). [12]

En esta etapa se lleva a cabo la elaboración de un modelo lógico-conceptual que incluye los objetos que se encuentren dentro de las actividades típicas que los usuarios realizan al emplear la aplicación, tratando de en lo posible omitir aspectos navegacionales, sino más bien basándose en los casos de uso obtenidos durante la etapa anterior.

2.2.3. Diseño Navegacional

La etapa del diseño navegacional se llega a conocer como están enlazadas las páginas por lo que se realiza un diagrama de nodos y enlaces. [11]

Dentro de esta etapa se obtiene tanto el modelo del espacio de navegación, el cual permite especificar los objetos a ser visitados por el navegador en la aplicación, así como el modelo que muestra la estructura que va a tener la aplicación para su navegación, en el que se establece como se relacionan dichos objetos.

2.2.4. Diseño de Presentación

Se realiza la presentación de la información que no fue especificada dentro del modelo navegacional como son las clases de navegación y de proceso que pertenecen a una página web. [11]

En esta etapa se describe la manera en la que los objetos determinados en la etapa de diseño navegacional serán presentados al usuario, es decir una manera ordenada de los objetos visibles para el usuario.

CAPÍTULO III.
DESARROLLO DE LA METODOLOGIA

3.1. Análisis de Requerimientos

3.1.1. Entrevista para análisis de requerimientos

Con la finalidad de conocer el funcionamiento actual de la empresa y detallar los requerimientos, se hizo una entrevista a la administradora de la empresa, con un listado de preguntas, de las cuáles se obtuvieron las siguientes repuestas:

1. **¿Cuántas personas trabajan dentro de la empresa?**

   La empresa actualmente cuenta con dos personas encargadas de llevar a cabo los procesos correspondientes de cada cliente que desee hacer sus declaraciones, la administradora y una asistente.

2. **¿En promedio, cuántas declaraciones se realizan al día?**

   Dependiendo la fecha de pago de cada contribuyente, en promedio se llevan a cabo alrededor de 15 a 20 declaraciones diarias.

3. **¿Cuál es el proceso de elaboración de una declaración?**

   El proceso se lo realiza de diferentes formas de acuerdo al tipo de contribuyente y la declaración que este requiera, por ejemplo el tipo de contribuyente es: Obligado a Llevar Contabilidad, y la declaración es: IVA Mensual, el cliente debe traer sus comprobantes de Compras y Ventas correspondientes al mes que desea declarar, además de los comprobantes de retención que se hayan registrado en dicho periodo.

4. **¿Para qué tipos de contribuyentes trabaja su empresa?**

Actualmente la empresa cuenta con contribuyentes obligados y no obligados a llevar contabilidad.

5. **¿Qué tipos de declaraciones realiza cada tipo de contribuyente?**

   Obligados a llevar contabilidad:

   - Declaración IVA (Mensual).
   - Declaración de Retenciones (Mensual).
   - Anexos Transaccionales (Mensual).
   - Declaración Impuesto a la Renta (Anual).

   No obligados a llevar contabilidad:

   - Declaración IVA (Mensual).
   - Declaración Impuesto a la Renta (Anual).

6. **¿Lleva algún registro de los clientes y sus declaraciones?**

   Actualmente la empresa no cuenta con ningún tipo de registro de clientes ni sus declaraciones, por lo que únicamente la empresa se basa en la fecha de declaración para constatar los clientes que deben realizar dicho proceso.

7. **¿Cuenta con algún método de comunicación constante con los clientes para el control de fechas y declaraciones?**

   No, cada cliente está pendiente de sus fechas de declaración, ocasionando sus atrasos y multas.

8. **¿Estaría usted de acuerdo con mejorar el control de clientes y sus declaraciones?**

   Si, sería muy productivo tener un mejor control de los datos de los clientes y las declaraciones que estos llevan a cabo.

9. **¿Cree usted que un sistema que automatice el proceso de comunicación y control de los clientes brindaría un beneficio para su empresa?**

   Si, puesto que brindaría una forma más productiva de llevar a cabo estos procesos que actualmente la empresa no los realiza y se reduciría la dependencia para con el contribuyente al no tener una forma constante de comunicación.

10. **Numere los procesos que se debería automatizar para mejorar la atención al cliente**

    - Registro de Clientes
    - Registro de Declaraciones
    - Comunicación con el Cliente

Resumen de la Entrevista

El análisis es de una empresa de asesoría tributaria para realizar declaraciones con respecto al SRI, la empresa cuenta con diferentes tipos de contribuyentes y a su vez estos realizan diferentes tipos de declaraciones, según lo especificado en la entrevista realizada a la administradora de la empresa, la comunicación con los clientes se presenta como un punto clave dentro del funcionamiento diario, la misma que actualmente no es eficiente razón principal para la implementación del proyecto de investigación.

En la entrevista se obtuvo un resumen de la manera actual en la que se realiza el proceso de declaraciones correspondiente a cada cliente, y se comprendió que si el cliente desea hacer su declaración se acerca a la

empresa donde estará la administradora o a su vez la auxiliar el cual comprueba el noveno digito de la cédula para determinar la fecha de la declaración correspondiente al contribuyente.

Dependiendo el tipo de declaración se pide los requisitos por ejemplo: Para la declaración mensual del IVA tomando en cuenta que la/el contribuyente es obligado a llevar contabilidad deberá entregar facturas egresos/compras, facturas de ingresos/ventas, retenciones esto dependerá del valor o monto de cada factura (Egresos/Compras), llenando con estos datos el formulario correspondiente enviándolo al sistema del SRI. Tomar en cuenta que para cada tipo de contribuyente existen algunos cambios en el proceso.

En el caso de que la/el contribuyente haya pasado la fecha de la declaración la misma se genera con valor agregado correspondiente a la multa, la misma que debe ser cancelada en una entidad bancaria, tras el resumen obtenido se entendió que la comunicación con los clientes previo a la declaración es muy deficiente, puesto que, se dependía de que el cliente se acerque a la empresa para poder realizar dicho proceso y que estos a su vez no tienen el suficiente conocimiento de sus fechas límites para realizar sus declaraciones exponiéndose a multas o pagos extras por retraso.

3.1.2. Documento de Requerimientos

3.1.2.1. Introducción

3.1.2.1.1. Propósito

El sistema es una aplicación web orientada a manejar los procesos, de control de clientes, que actualmente no son realizados dentro de la empresa.

3.1.2.1.2. Visión General

En el proyecto de investigación constarán, tanto los requerimientos funcionales de la aplicación web, así como los casos de uso del sistema.

3.1.2.1.3. Descripción General

En la entrevista realizada a la administradora de la empresa de Asesoría Tributaria y Contable MD, la aplicación permitirá a sus usuarios realizar el registro de los datos de sus clientes y sus declaraciones. De la misma forma se implementará un sistema automático de mails para notificar las fechas de pago y las declaraciones realizadas, a los clientes. A esta aplicación web tendrán acceso en todo momento la administradora y los clientes, siempre que tengan acceso a internet.

3.1.2.2. Requerimientos Específicos

3.1.2.2.1. Requerimientos Funcionales

Registro de Contribuyentes

El aplicativo web deberá permitir llevar a cabo el registro completo de los contribuyentes que van a realizar sus declaraciones en la empresa, estos datos son:

- Ruc
- Razón Social
- Correo Electrónico
- Teléfono Casa
- Teléfono Móvil
- Tipo de Contribuyente
- Obligado a llevar Contabilidad

Registro de Declaraciones

El sistema web será capaz de registrar las declaraciones realizadas por cada contribuyente, indicando los siguientes datos:

- Fecha de Registro
- Cliente
- Tipo de Declaración
- Período
- Mes
- Valor Pagado
- Multa

Registro de Usuarios

.El aplicativo web contará con un módulo capaz de llevar a cabo el registro completo de los usuarios que van a tener contacto con el sistema, estos datos son:

- Ruc

- Razón Social
- Contraseña
- Perfil
- Estado

Registro de Tipos de Declaración

El sistema podrá realizar el registro de los tipos de declaración correspondientes a cada contribuyente y el período en el que estás deben ser elaboradas.

3.1.3. Diagramas Casos de Uso

3.1.3.1. Contribuyente Obligado Llevar Contabilidad

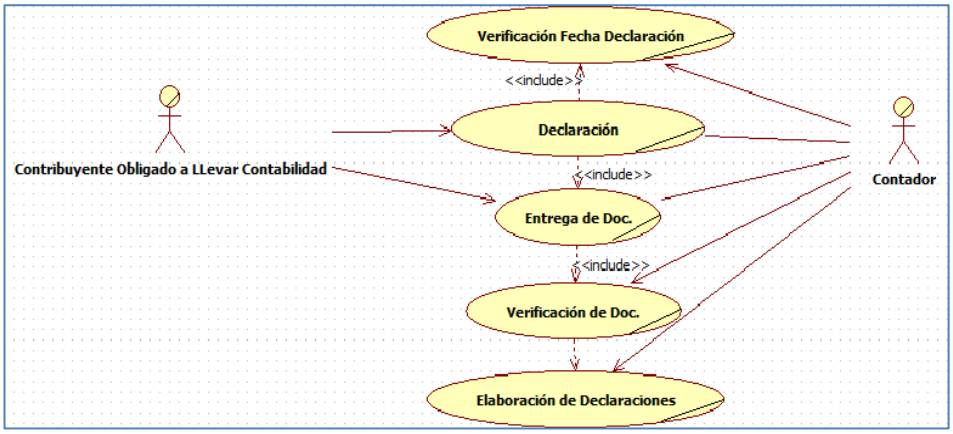


Ilustración 3.1. Contribuyente Obligado a Llevar

Elaborado por: Luis Torres

3.1.3.2. Generalización Declaración

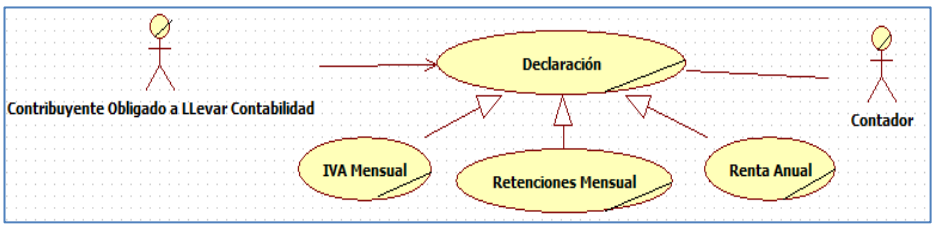


Ilustración 3.2. Generalización Declaración

Elaborado por: Luis Torres

3.1.3.3. Generalización de Documentos

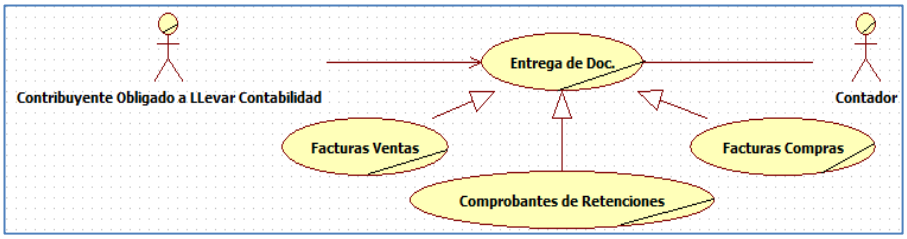


Ilustración 3.3. Generalización de Documentos

Elaborado por: Luis Torres

3.1.3.4. Generalización Elaboración de Declaraciones

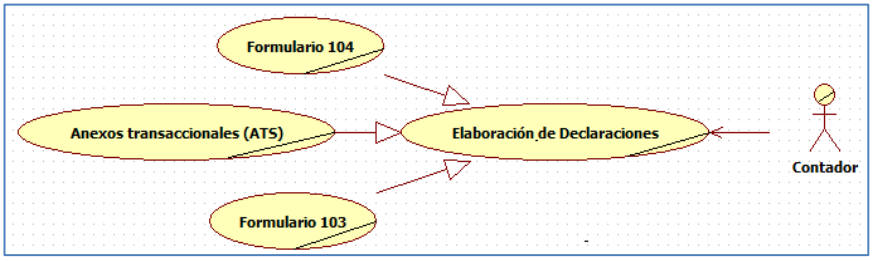


Ilustración 3.4. Generalización Elaboración de Declaraciones

Elaborado por: Luis Torres

3.1.3.5. Resumen Contribuyente Obligado a Llevar Contabilidad

El contribuyente Obligado a llevar contabilidad debe realizar mensualmente la declaración de IVA y Retenciones o anualmente la declaración Renta con el contador las declaraciones en la fecha que corresponde al noveno digito del RUC, lo que evitara pagar multas e intereses en caso de atraso, para lo cual mensualmente deberá el cliente entregar las facturas emitidas (ventas), recibidas (compras) y retenciones del mes pasado. Para que el contador una vez recibido la documentación verifique la validez y proceda a registrar en los libros correspondientes y pueda elaborar los Formulario 104 IVA, Formulario 103 Retenciones.

3.1.3.6. Contribuyente No Obligado a Llevar Contabilidad

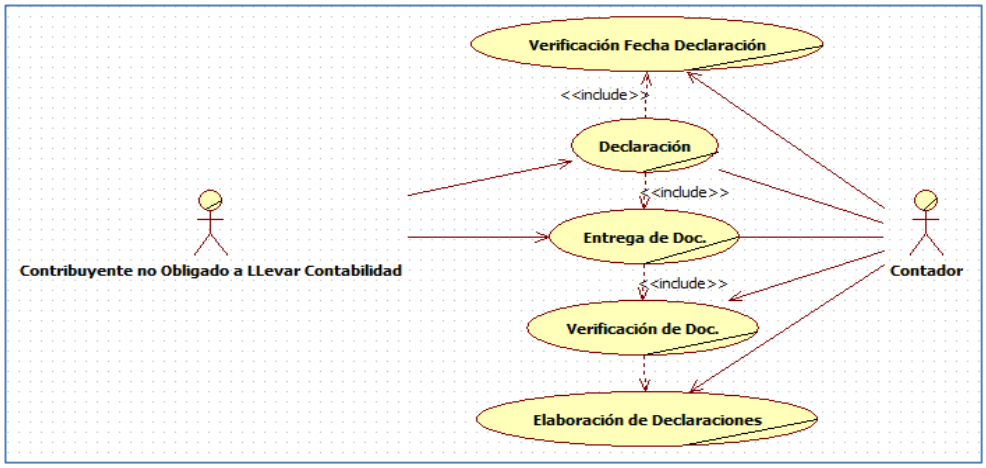


Ilustración 3.5. Contribuyente no Obligado a llevar Contabilidad

Elaborado por: Luis Torres

3.1.3.7. Generalización Declaración

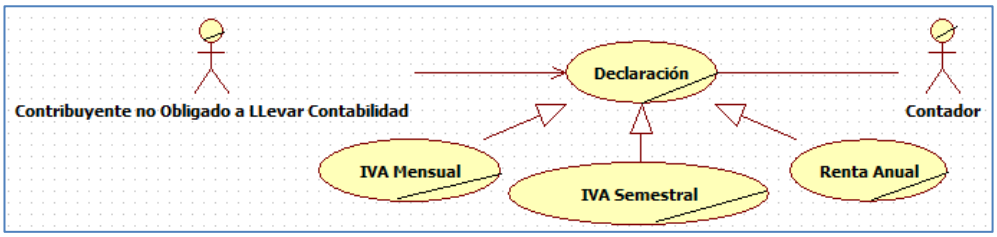


Ilustración 3.6. Generalización Declaración

Elaborado por: Luis Torres

3.1.3.8. Generalización Entrega de Documentos

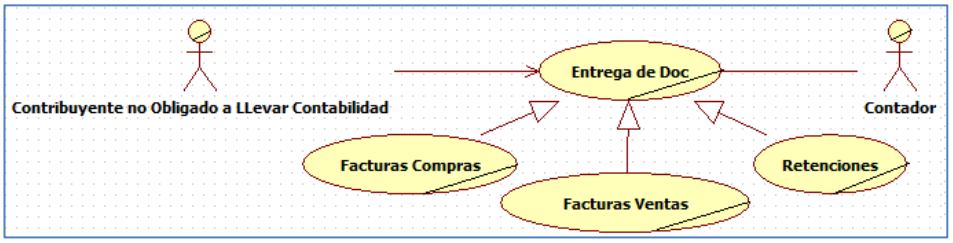


Ilustración 3.7. Generalización Entrega de Documentos

Elaborado por: Luis Torres

3.1.3.9. Elaboración Declaraciones



Ilustración 3.8. Elaboración Declaraciones

Elaborado por: Luis Torres

3.1.3.10. Resumen

El contribuyente no Obligado a llevar contabilidad puede realizar mensual o semestral la declaración de IVA y anualmente la declaración Renta con el contador las declaraciones en la fecha que corresponde al noveno digito del RUC, lo que evitara pagar multas e intereses en caso de atraso, para lo cual debe el cliente entregar las facturas emitidas (ventas), recibidas (compras) y retenciones que le recibe el cliente para la declaración mensual, semestral y anual. Para que el contador una vez recibido la documentación verifique la validez y proceda a registrar en los libros correspondientes y pueda elaborar el Formulario 104 no Obligado a llevar Contabilidad.

3.1.3.11. IVA por Pagar Fuente

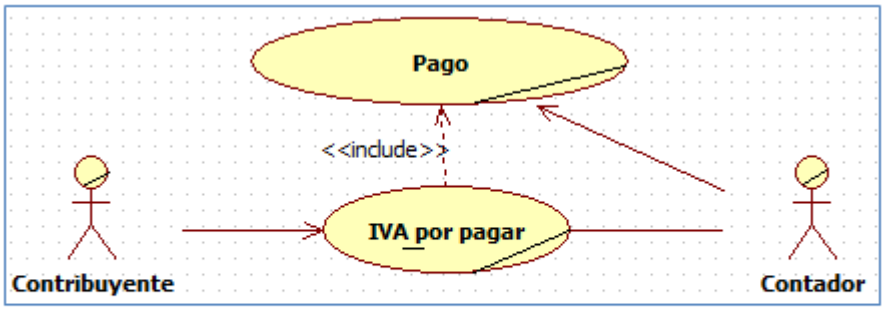


Ilustración 3.9. IVA por Pagar

Elaborado por: Luis Torres

3.1.3.12. Resumen

En el caso de haber IVA por pagar deberá acercarse al banco a cancelar la cantidad, caso contrario como resultado será una declaración valor en cero cuando existe crédito tributario.

3.1.4. Diagramas Casos de Uso del Sistema

3.1.4.1. Registro de Cliente



Ilustración 3.10. Registro de Cliente

Elaborado por: Luis Torres

Documento de Casos de Uso del Sistema

1. Nombre: Registro Cliente.

2. Actores: Cliente, Administrador.

3. Propósito: Registro de los datos del cliente.

4. Resumen: El cliente acude a solicitar la elaboración de una declaración para cumplir con lo demandado por el SRI en una fecha específica, la administradora verifica la fecha para la declaración.

5. Curso de los eventos:

Actores	Sistema
1. La administradora pide al cliente información para el registro en el sistema.	
2. El cliente le da los datos necesario para el registro.	
	3. En el sistema se ingresa los datos necesarios del cliente y guarda.
	4. El sistema extraerá el noveno digito del RUC para un envío masivo de mail hacia los clientes indicando el día que deberá realizar las declaraciones.

6. Cursos Alternos:

- Se pasó de la fecha para la declaración, esto genera multa.

3.1.4.2. Administración Fechas

Documento de Casos de Uso del Sistema

1. Nombre: Administración Fechas.

2. Actores: Sistema.

3. Propósito: Genera la fecha.

4. Resumen: El cliente acude a solicitar la elaboración de una declaración para cumplir con lo demandado por el SRI en una fecha específica, la administradora verifica la fecha para la declaración.

5. Curso de los eventos:

Actores	Sistema
1. La administradora pide al cliente información para el registro en el sistema.	
2. El cliente le da los datos necesario para el registro.	
	3. En el sistema se ingresa los datos necesarios del cliente y guarda.

6. Cursos Alternos:

- Se pasó de la fecha para la declaración, esto genera multa.

3.1.3.3. Registro Declaraciones

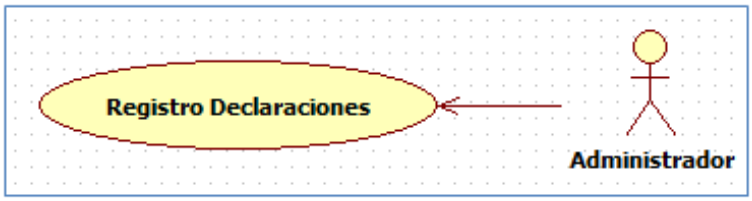


Ilustración 3.11. Registro Declaración

Elaborado por: Luis Torres

Documento de Casos de Uso del Sistema

1. Nombre: Registro Declaraciones.

2. Actores: Administrador.

3. Propósito: Registro de los datos de las declaraciones.

4. Resumen: El cliente acude a solicitar la elaboración de una declaración para cumplir con lo demandado por el SRI en una fecha específica, la contadora realiza el respectivo trámite para el contribuyente una vez generado los formularios respectivos la administradora registra en el sistema los datos de las declaraciones realizadas para en envió al cliente mediante un mail.

5. Curso de los eventos:

Actores	Sistema
1. El cliente pide al contador trámite declaraciones para el registro en el sistema SRI.	
2. El cliente entrega los documentos necesarios para el cálculo y generar los formularios respectivos	

necesarios para el registro.	
	3. En el sistema se ingresa los datos necesarios de las declaraciones del cliente guarda y envío de mail.

6. Cursos Alternos:

Mail no enviado o email incorrecto del cliente.

3.1.3.4. Envío de Mail



Ilustración 3.12. Envío de Mail

Elaborado por: Luis Torres

Documento de Casos de Uso del Sistema

1. Nombre: Envío de Mail.

2. Actores: Administrador, Cliente.

3. Propósito: Envío de mail de información para el cliente.

4. Resumen: La administradora una vez registrada la información de la declaración guarda y se envía instantáneamente un mail al cliente, y a su

vez el sistema le enviará automáticamente la fecha siguiente para la declaración correspondiente.

5. Curso de los eventos:

Actores	Sistema
1. La administradora registra la declaración y da clic en Guardar.	
	2. Una vez que guarda la información hace el envío instantáneo mail al cliente con la información de la declaración registrada.
	3. También hará el envío mail al cliente de la fecha siguiente a declarar lo realizara automáticamente.

6. Cursos Alternos:

Mail no enviado o email incorrecto del cliente.

3.2. Fase II Diseño Conceptual

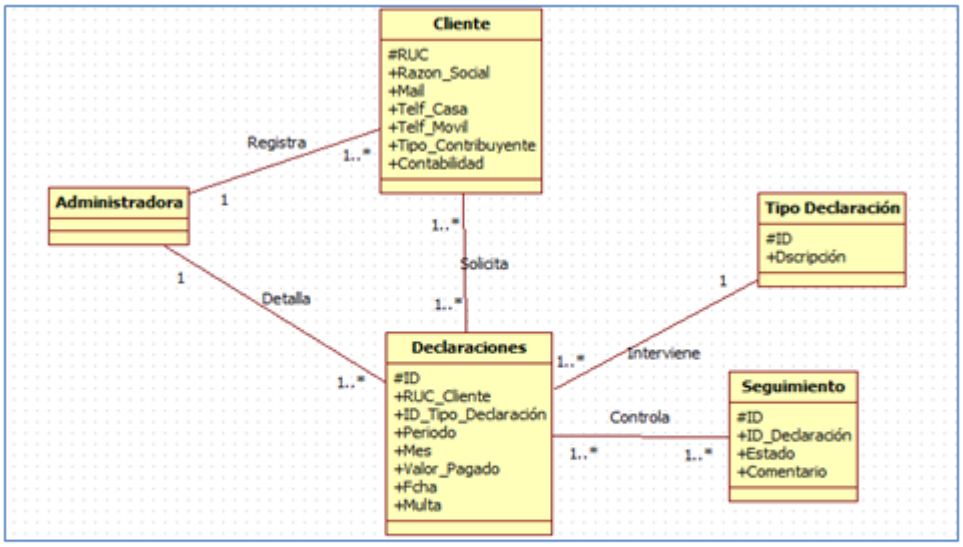


Ilustración 3.13. Diseño Conceptual

Elaborado por: Luis Torres

3.3. Modelo de Datos (Diseño de Datos)

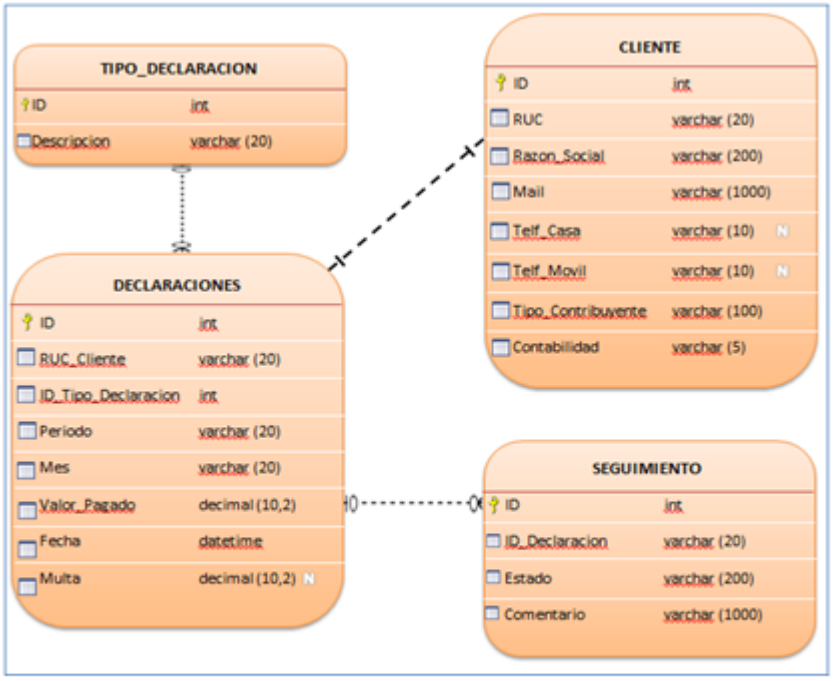


Ilustración 3.14. Modelo de Datos

Elaborado por: Luis Torres

3.3.1. Diccionario de Datos

Tablas Principales

TABLAS	DESCRIPCION
CLIENTE	Almacena la información de los contribuyentes
TIPO DECLARACION	Almacena la información de los tipos de declaraciones existentes
DECLARACION	Almacena la información de las declaraciones realizadas por cada contribuyentes
SEGUIMIENTO	Almacena la información de los estados en los que se encuentran las declaraciones de los contribuyentes

Elaborado por: Luis Torres

3.3.2. Tablas Específicas

Tabla Cliente

Campo	Tipo	Rango	Nulo	Clave Primaria	Clave Foránea	Descripción
ID	Int		NO	SI	NO	Clave primaria del cliente
RUC	varchar	20	NO	NO	NO	RUC del cliente
Razon_Social	varchar	200	NO	NO	NO	Razón Social del cliente
Mail	varchar	1000	NO	NO	NO	Mail del cliente
Telf_Casa	varchar	10	SI	NO	NO	Teléfono convencional del cliente
Telf_Movil	varchar	10	SI	NO	NO	Teléfono móvil del cliente
Tipo_Contribuyente	varchar	100	NO	NO	NO	Tipo de Contribuyente del cliente
Contabilidad	varchar	5	NO	NO	NO	Obligado o no a llevar contabilidad

Elaborado por: Luis Torres

Tabla Declaraciones

Campo	Tipo	Rango	Nulo	Clave Primaria	Clave Foránea	Descripción
ID	Int		NO	SI	NO	Clave primaria de declaraciones
RUC_Cliente	varchar	20	NO	NO	SI	Se obtiene la clave foránea del cliente
ID_Tipo_Declaracion	Int		NO	NO	SI	Se obtiene la clave foránea del tipo de declaración
Periodo	varchar		NO	NO	NO	Año al que corresponde la declaración
Mes	varchar	20	NO	NO	NO	Mes al que corresponde la declaración
Valor_Pagado	decimal	10,2	NO	NO	NO	Valor Cancelado de la declaración
Fecha	datetime		NO	NO	NO	Fecha en que se realizó la declaración
Multa	decimal	10,2	SI	NO	NO	Multa de la declaración en caso de tenerla

Elaborado por: Luis Torres

Tabla Tipo de Declaración

Campo	Tipo	Rango	Nulo	Clave Primaria	Clave Foránea	Descripción
ID	Int		NO	SI	NO	Clave primaria de tipos de declaración
Descripcion	varchar	20	NO	NO	SI	Nombre del tipo de declaración

Elaborado por: Luis Torres

Tabla Seguimiento

Campo	Tipo	Rango	Nulo	Clave Primaria	Clave Foránea	Descripción
ID	Int		NO	SI	NO	Clave primaria de tipos de declaración
ID_Declaracion	varchar	20	NO	NO	SI	Se obtiene la clave foránea de declaración
Estado	varchar	200	NO	NO	NO	Estado en el que se encuentra la declaración
Comentario	varchar	1000	SI	NO	NO	Comentario sobre la declaración realizada

Elaborado por: Luis Torres

3.4. Diseño de Navegación

3.4.1. Diagrama de Componentes

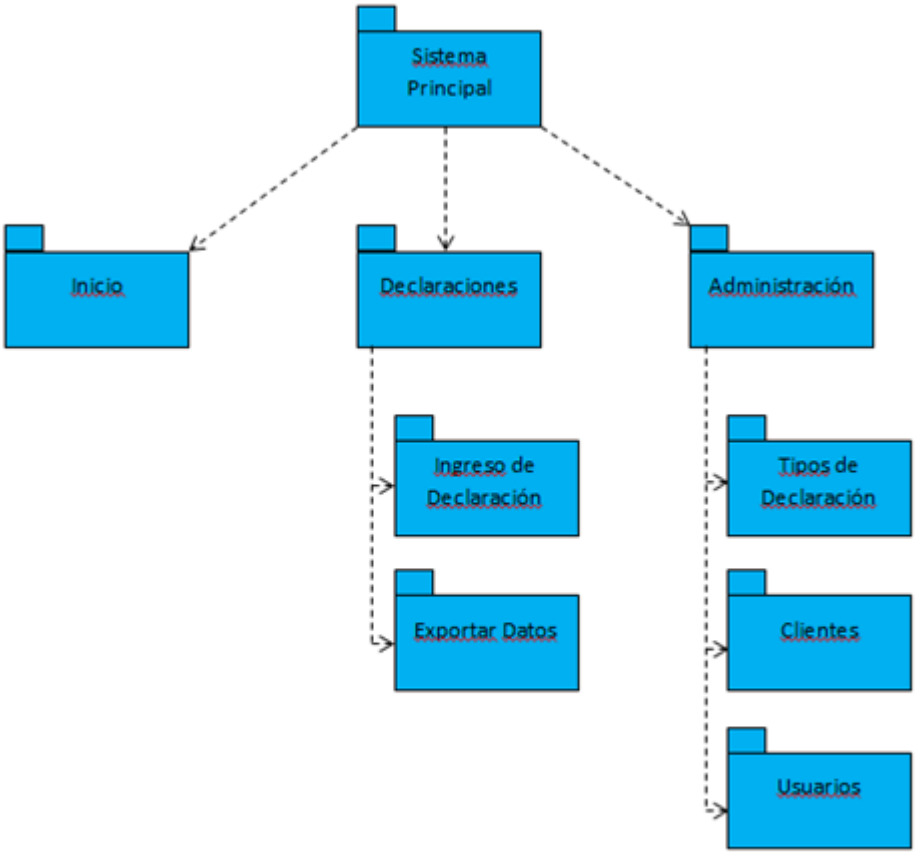


Ilustración 3.15. Diseño de Navegación

Elaborado por: Luis Torres

3.4.2. Modelo de Navegación

3.4.2.1. Simbología Utilizada

- Clase de Navegación
- Menú
- Pregunta
- Clase de Proceso

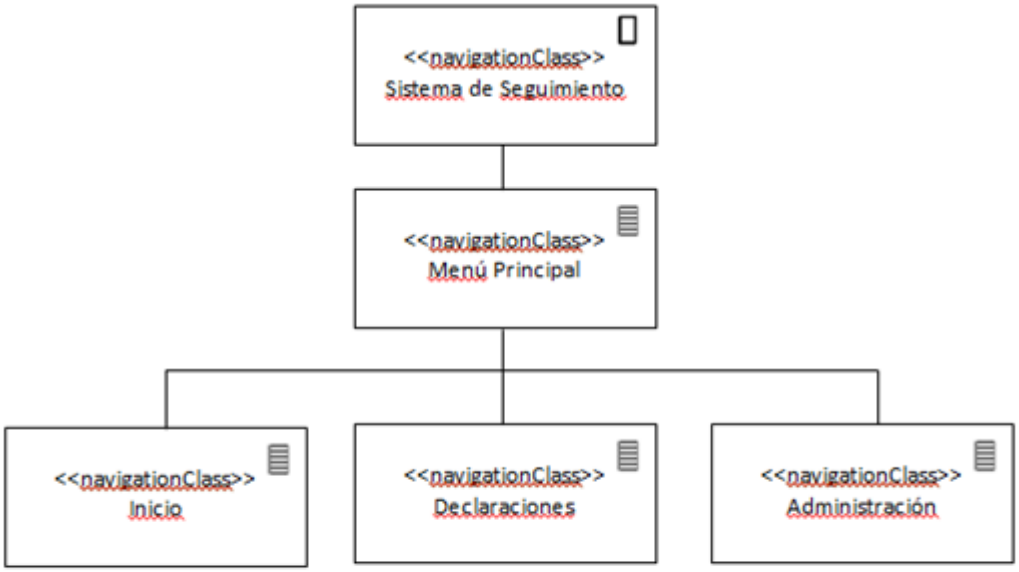


Ilustración 3.16.. Diagrama UWE Navegación Sistema Principal

Elaborado por: Luis Torres

3.4.2.2. Nuevo Cliente

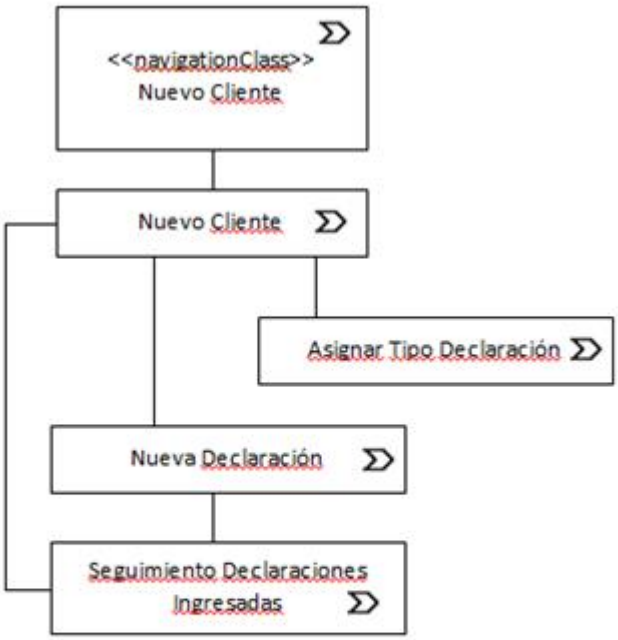


Ilustración 3.17. Diagrama UWE Nuevo Cliente

Elaborado por: Luis Torres

3.4.2.3. Cliente Existente

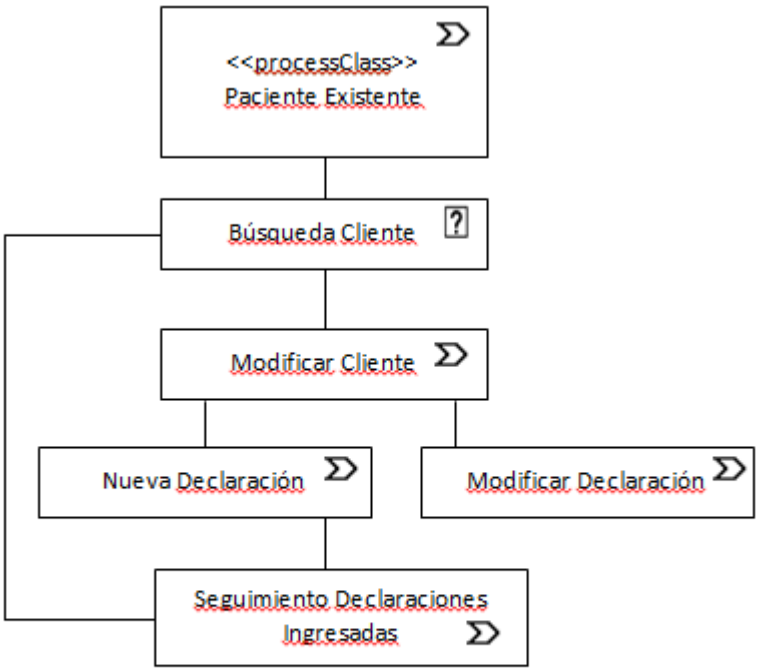


Ilustración 3.18. Diagrama UWE Paciente Existente

Elaborado por: Luis Torres

3.4.2.4. Administración

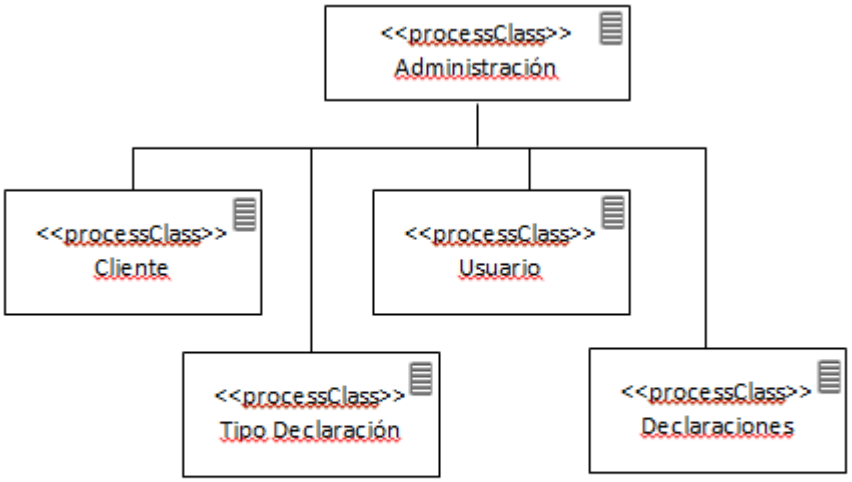


Ilustración 3.19. Diagrama UWE Administración

Elaborado por: Luis Torres

3.4.2.5. Declaraciones

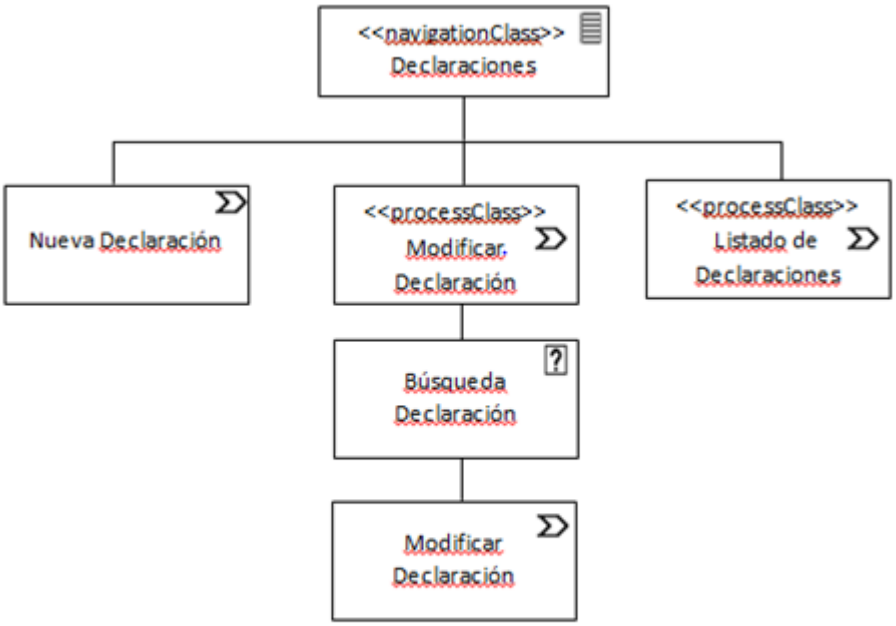


Ilustración 3.20. Diagrama UWE Declaraciones

Elaborado por: Luis Torres

3.4.2.6. Tipos de Declaración

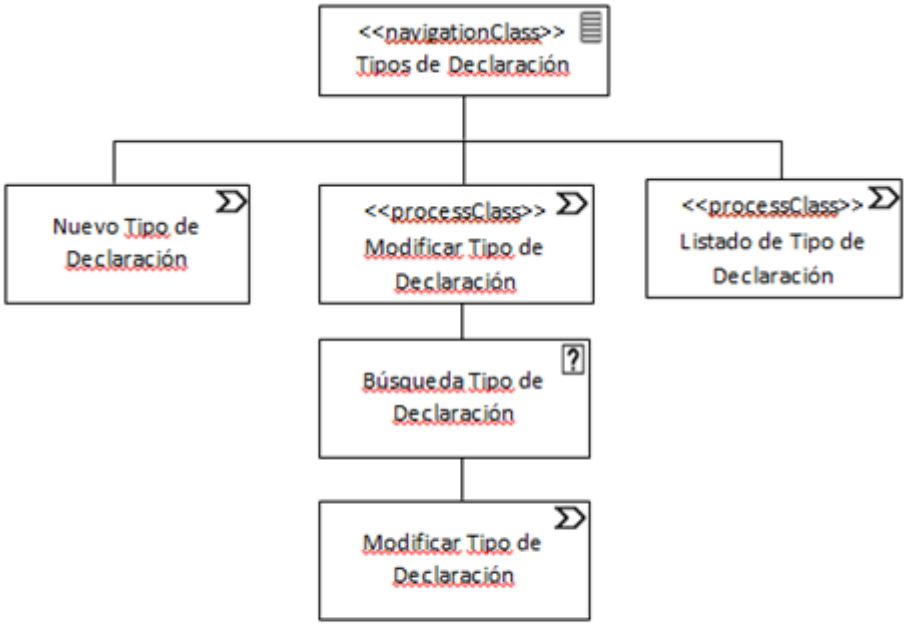


Ilustración 3.21. Diagrama UWE Tipo de Declaración

Elaborado por: Luis Torres

3.4.2.7. Usuarios

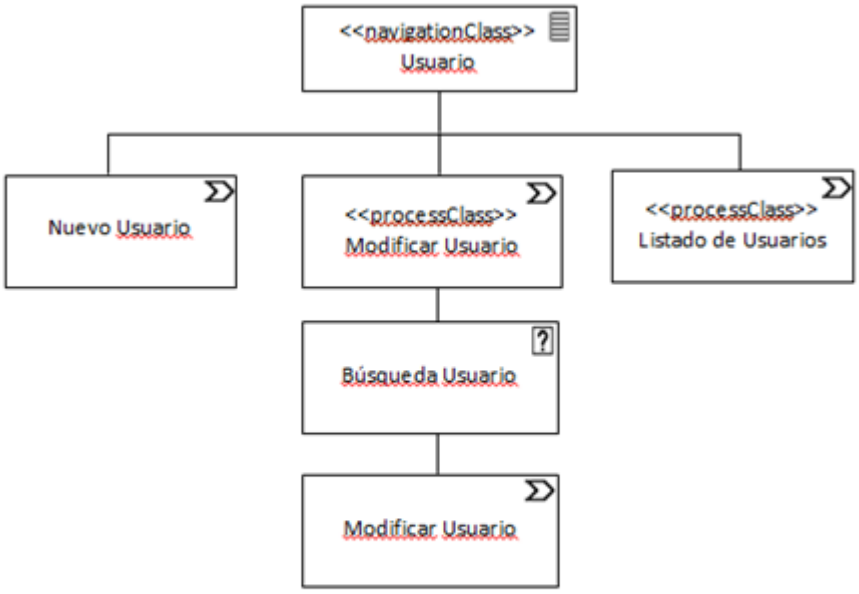


Ilustración 3.22. Diagrama UWE Usuario

Elaborado por: Luis Torres

3.4.2.8. Clientes

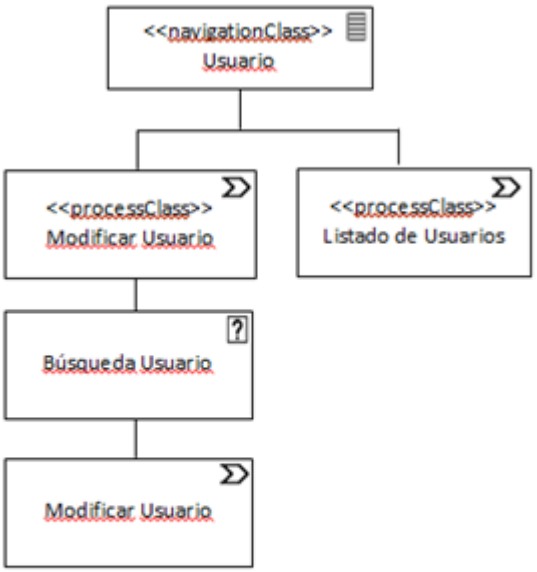


Ilustración 3.23. Diagrama UWE Usuario

Elaborado por: Luis Torres

3.5. Modelo de Presentación

3.5.1. Diagrama de Presentación

Representación de Cada Símbolo

- Texto
- Botón
- Formulario
- Selección
- Entrada de Texto

3.5.1.1. Registro de un Cliente

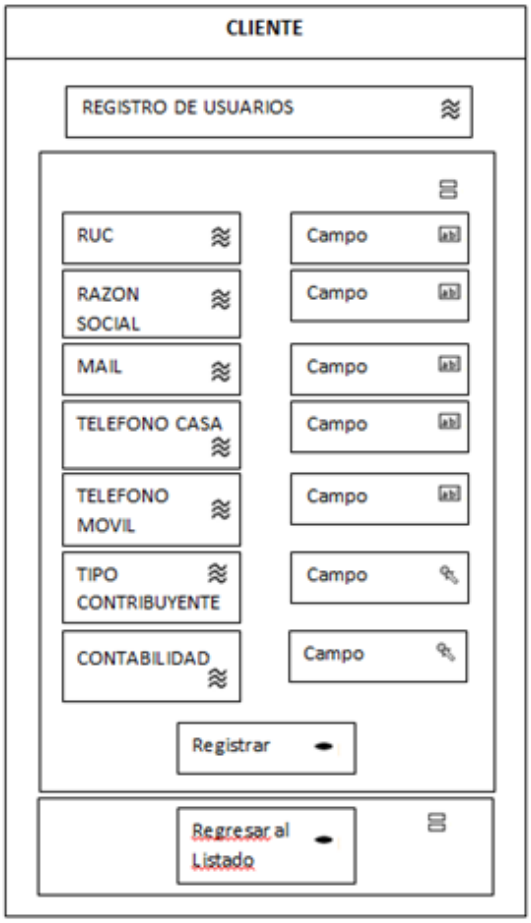


Ilustración 3.24. Diagrama UWE Formulario Cliente

Elaborado por: Luis Torres

3.5.1.2. Registro de un Usuario

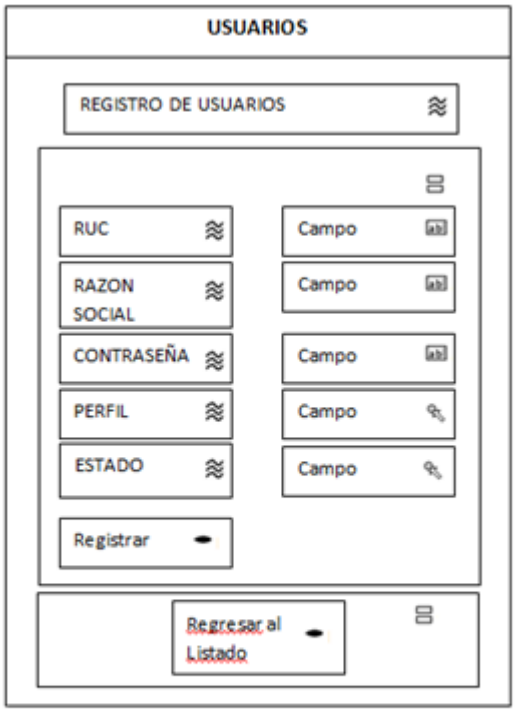


Ilustración 3.25. Diagrama UWE Formulario Usuario

Elaborado por: Luis Torres

3.5.1.3. Registro de Tipo de Declaración



Ilustración 3.26. Diagrama UWE Formulario Tipo de Declaración

Elaborado por: Luis Torres

3.1.5.4. Registro de Declaración

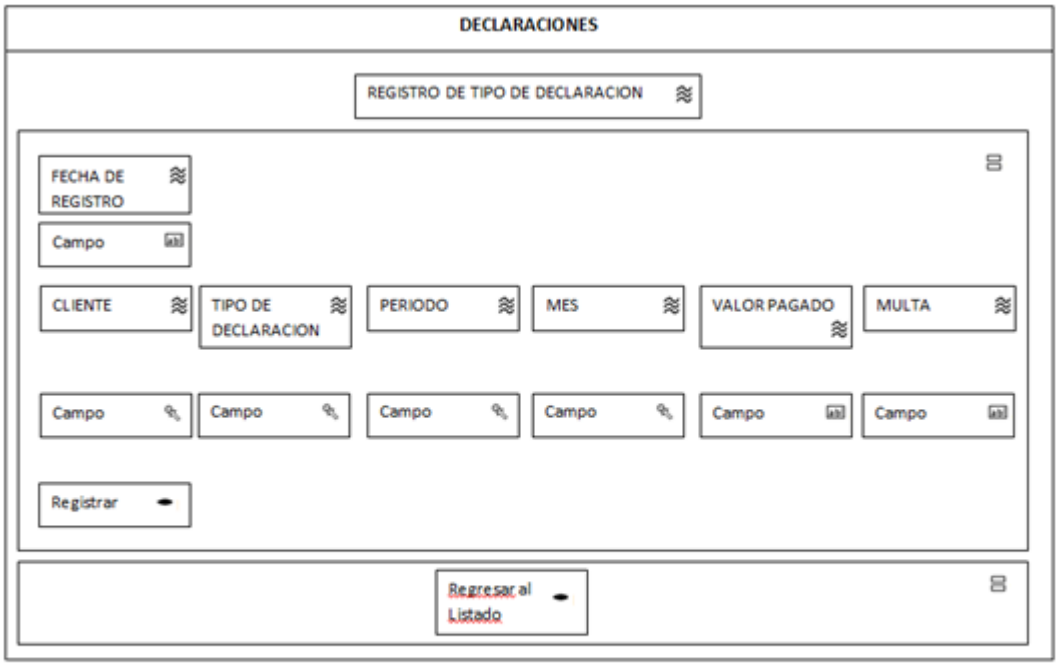


Ilustración 3.27 Diagrama UWE Formulario Declaraciones

Elaborado por: Luis Torres

CAPÍTULO IV.

RESULTADOS

4.1. Aplicación

4.1.1. Login

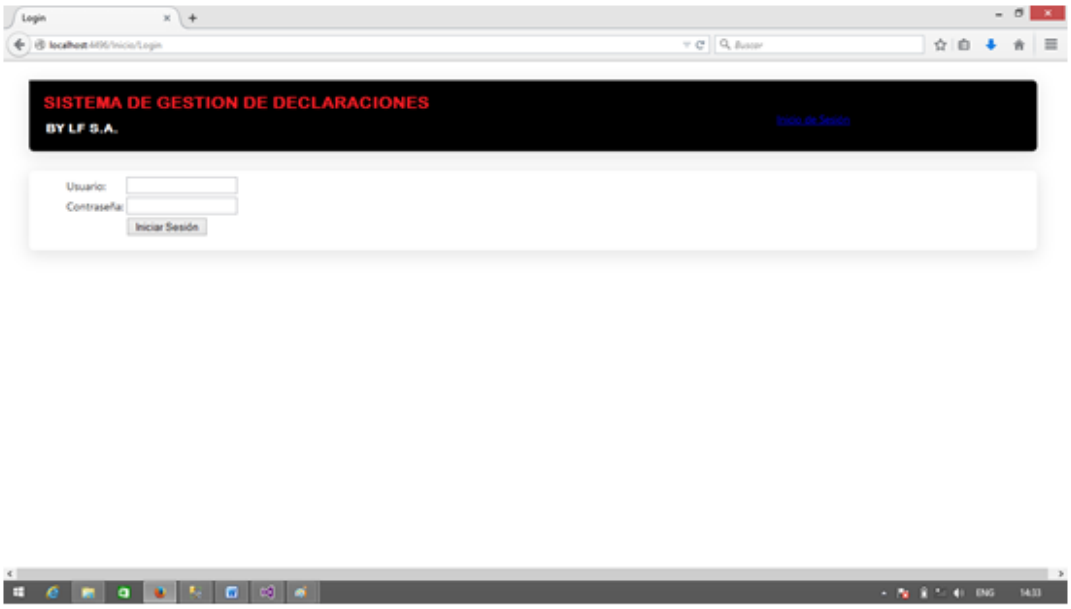


Ilustración 4.1.. Login

Elaborado por: Luis Torres

4.1.2. Página Principal Perfil Administrador

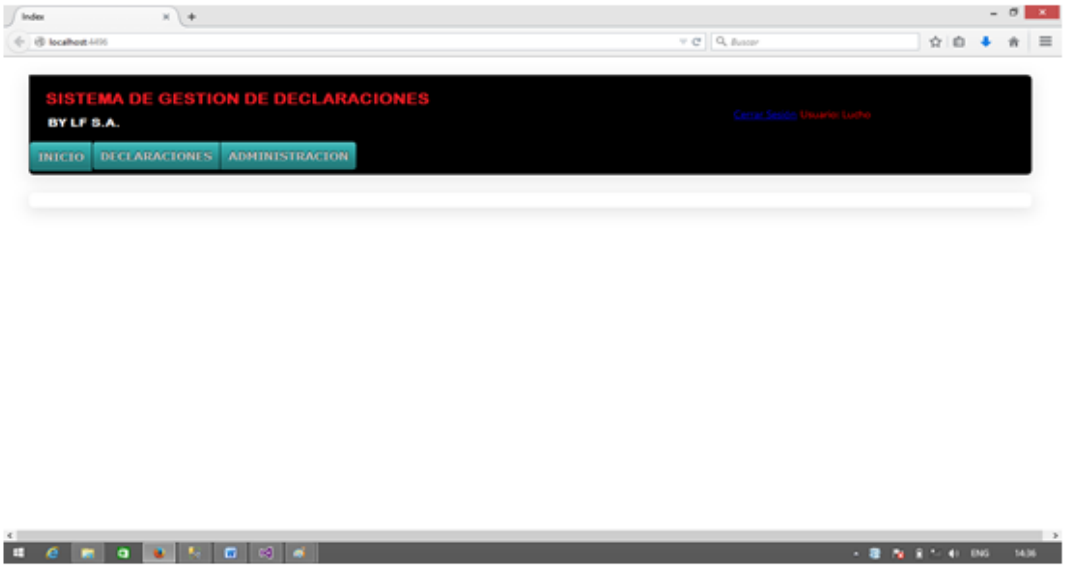


Ilustración 4.2. Página Principal

Elaborado por: Luis Torres

4.1.3. Página Principal Perfil Usuario

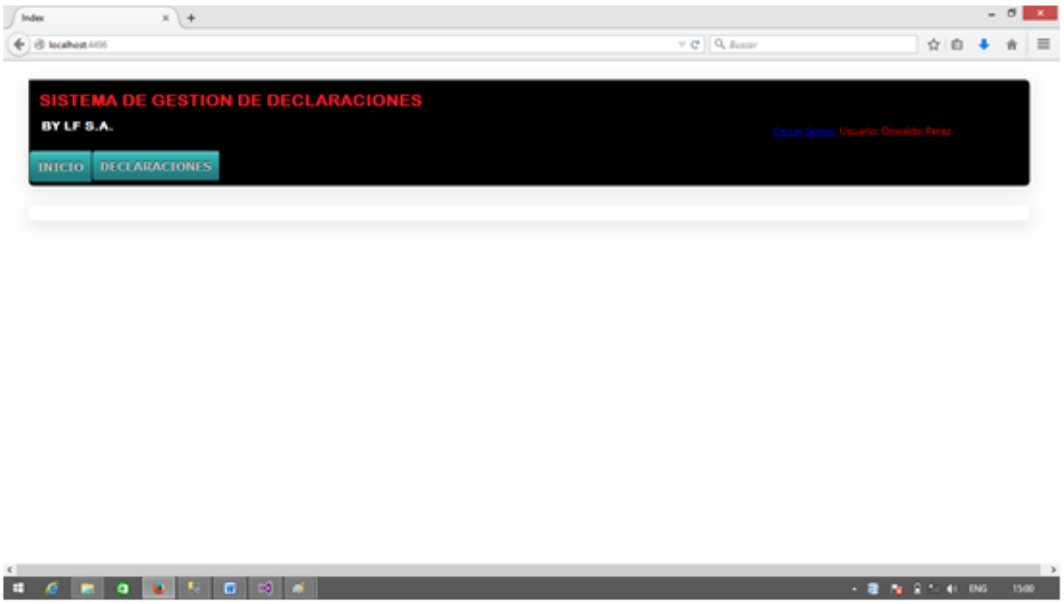


Ilustración 4.3. Página Principal Perfil

Elaborado por: Luis Torres

4.1.4. Registro Usuarios

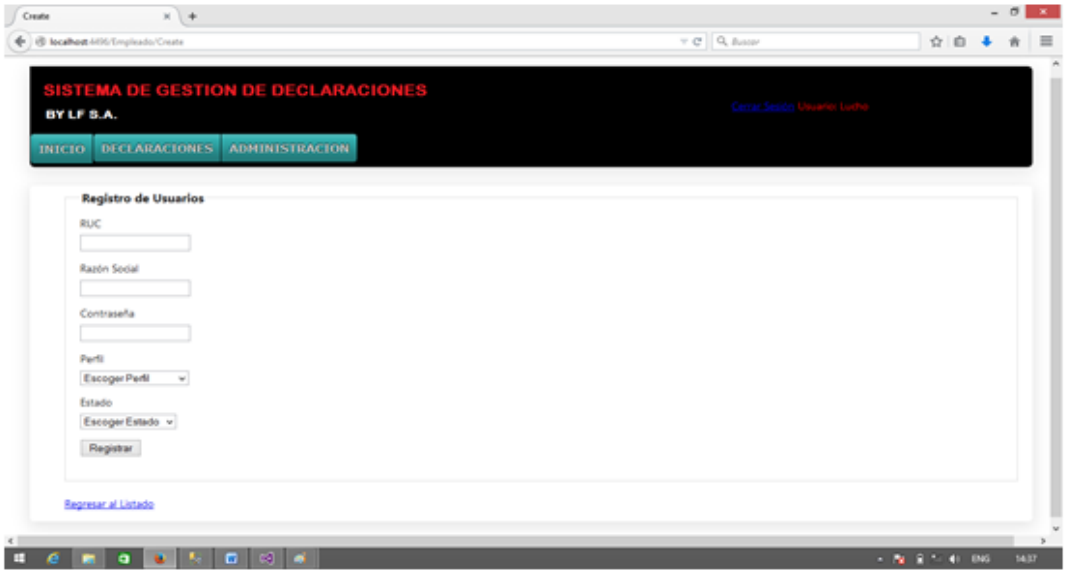


Ilustración 4.4. Registro Clientes

Elaborado por: Luis Torres

4.1.5. Listado Usuarios

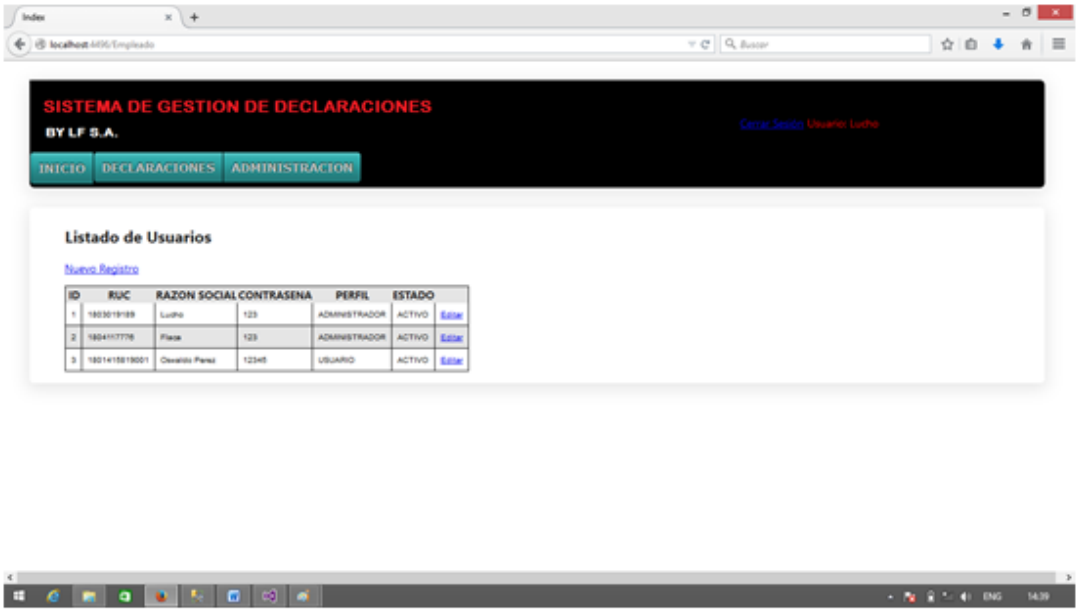


Ilustración 4.5. Listado Usuarios

Elaborado por: Luis Torres

4.1.6. Edición Usuarios

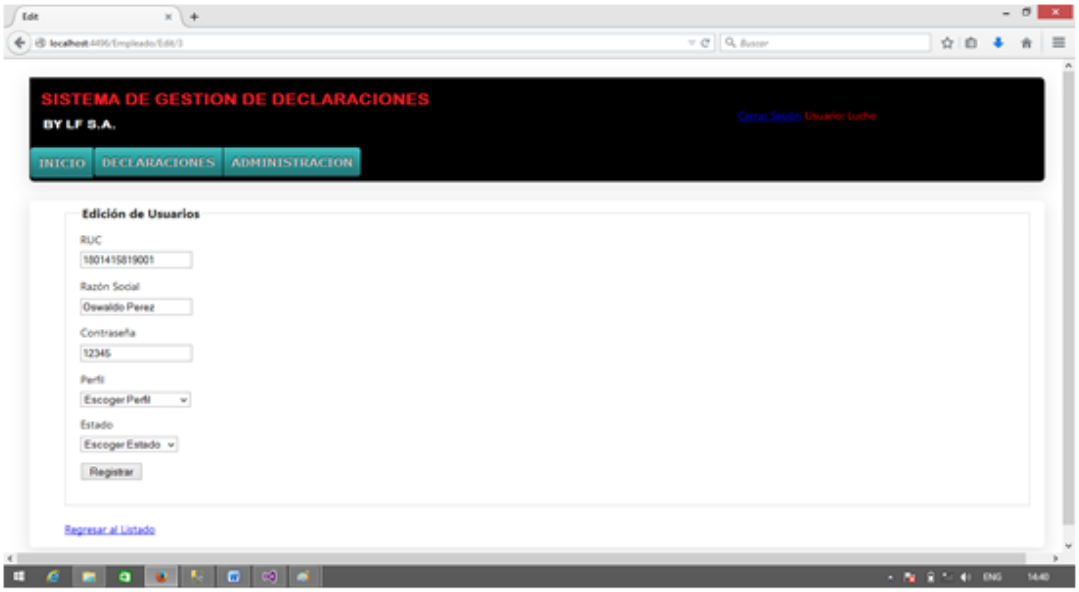


Ilustración 4.6. Edición Usuarios

Elaborado por: Luis Torres

4.1.7. Registro de Cliente

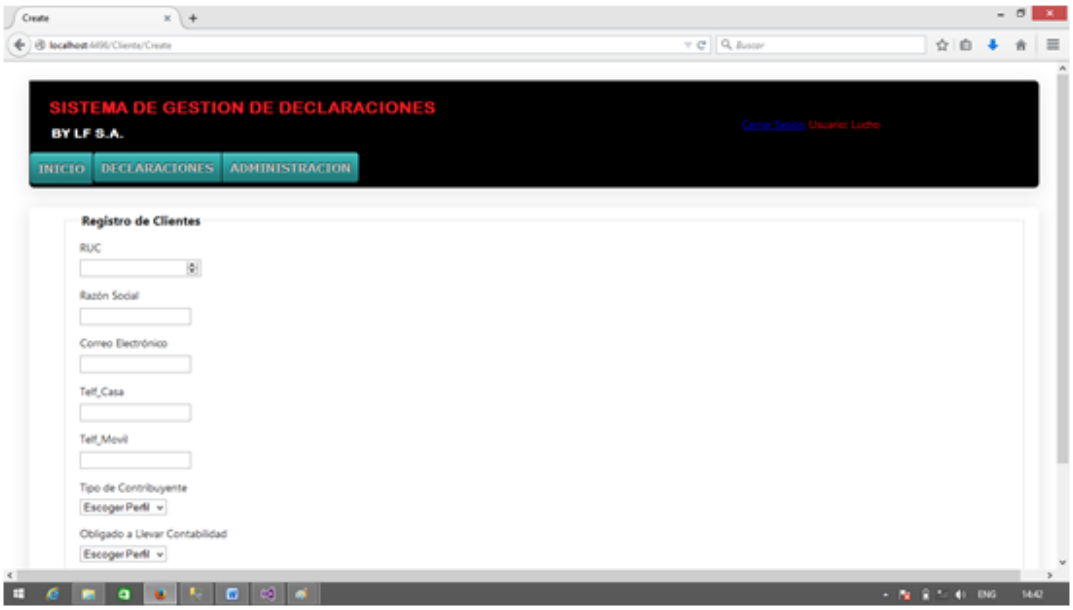


Ilustración 4.7. Registro de Cliente

Elaborado por: Luis Torres

4.1.8. Listado de Clientes

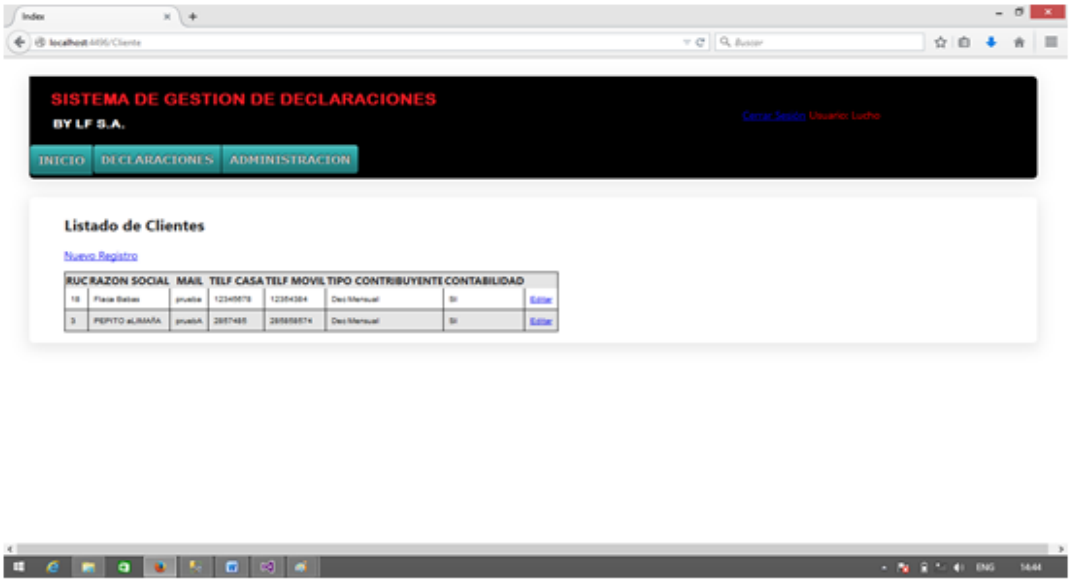


Ilustración 4.8. Listado de Clientes

Elaborado por: Luis Torres

4.1.9. Edición de Cliente

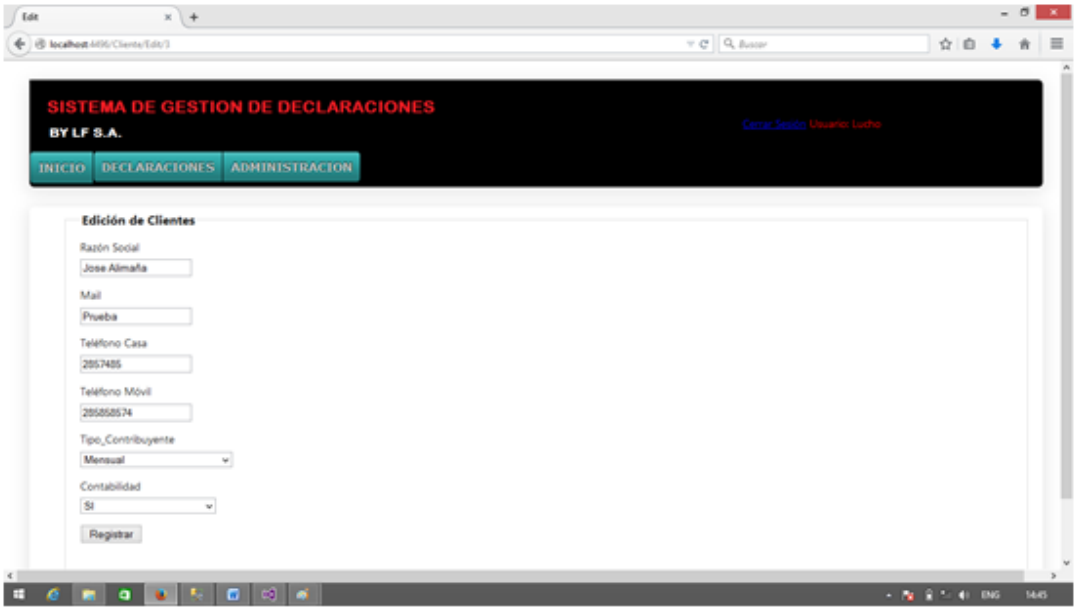


Ilustración 4.9. Edición de Cliente

Elaborado por: Luis Torres

4.1.10. Ingreso Tipo de Declaración

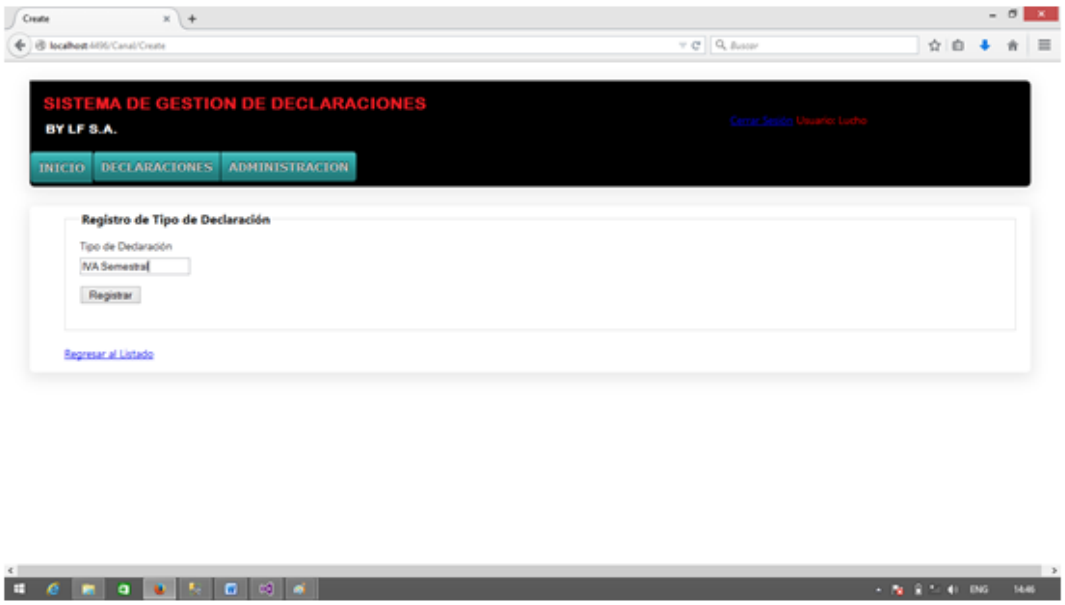


Ilustración 4.10. Tipo de Declaración

Elaborado por: Luis Torres

4.1.11. Listado Tipo de Declaración

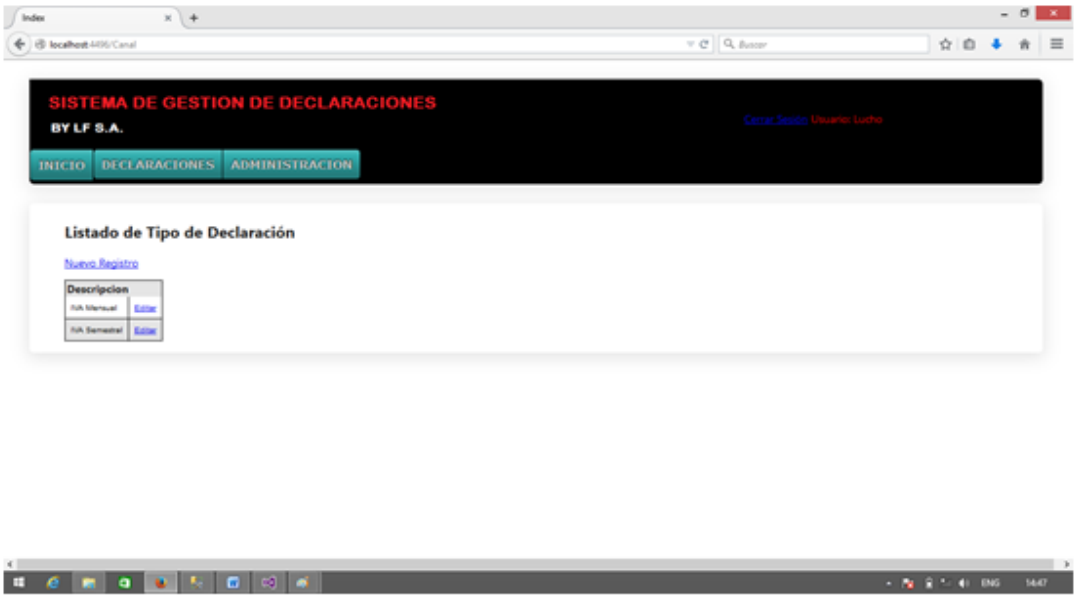


Ilustración 4.11. Listado Tipo de Declaración

Elaborado por: Luis Torres

4.1.12. Edición Tipo de Declaración

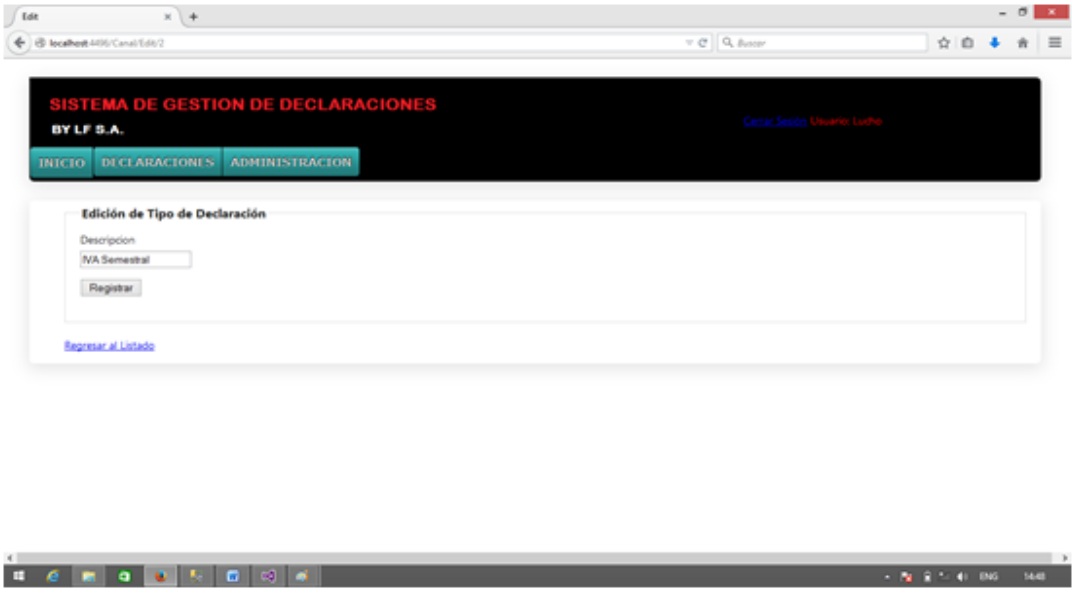


Ilustración 4.12. Edición Tipo de Declaración

Elaborado por: Luis Torres

4.1.13. Ingreso Declaración

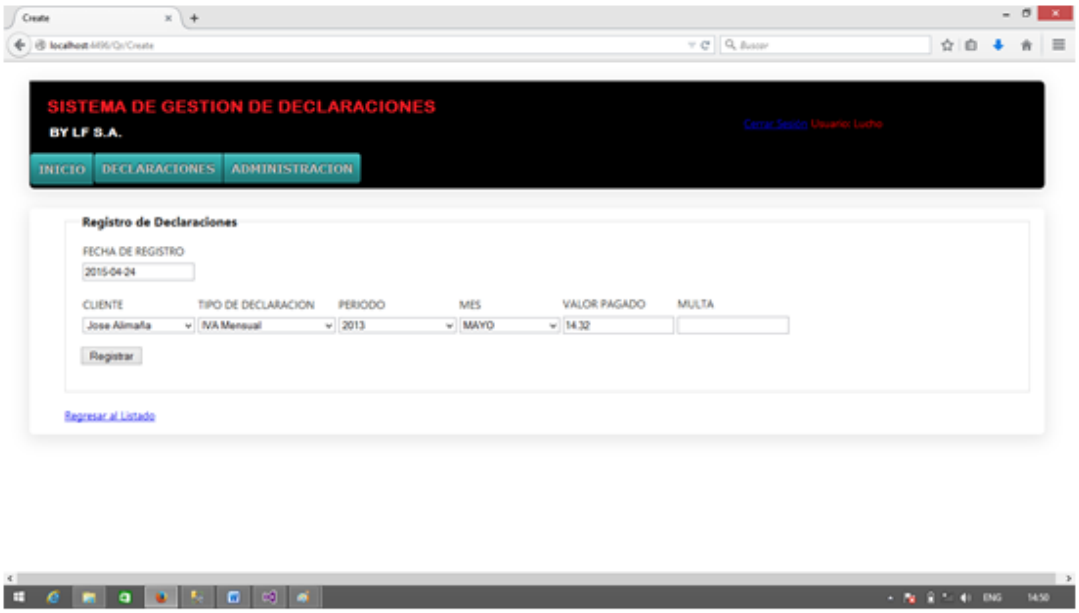


Ilustración 4.13. Registro de Declaración

Elaborado por: Luis Torres

4.1.14. Listado de Declaraciones

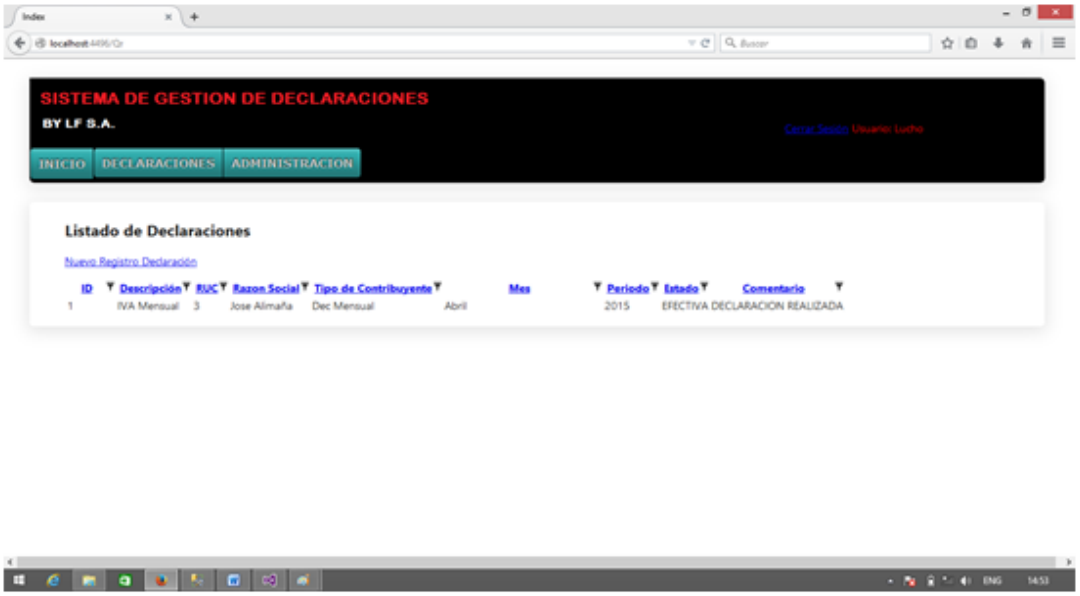


Ilustración 4.14. Listado de Declaraciones

Elaborado por: Luis Torres

4.1.15. Exportar Reporte de Declaraciones

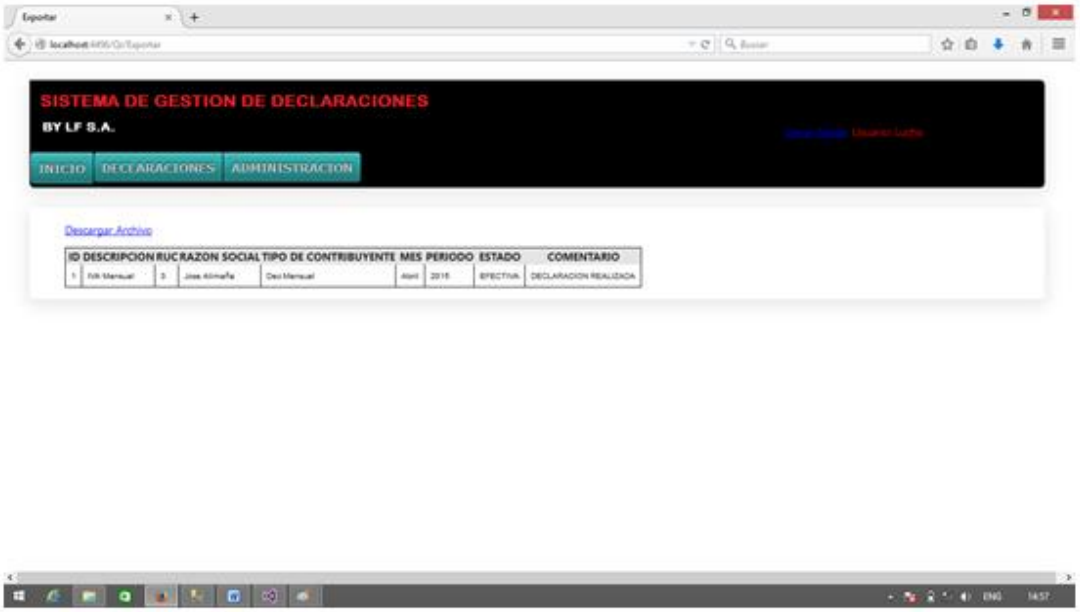


Ilustración 4.15. Reporte de Declaraciones

Elaborado por: Luis Torres

CAPITULO V.

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- De acuerdo con los requerimientos obtenidos y tras el desarrollo realizado, se pudo visualizar que el sistema brinda una mejor forma de llevar el control de las diferentes declaraciones que deben realizar cada contribuyente.
- Mediante el sistema de envíos automáticos de mails se ha podido observar una mejor comunicación entre los contribuyentes y la empresa.
- Los contribuyentes ahora están más informados acerca del estado de sus declaraciones, de las obligaciones pendientes que tienen, de sus fechas de declaración, de los valores declarados y en los casos en los que corresponda los valores de multa.
- La diferenciación entre perfiles (Administrador y Usuario), permite que la aplicación brinde acceso a las funciones correspondientes que deben tener cada uno de estos, evitando así que un usuario pueda manejar el sistema de forma incorrecta y brindándole todos los privilegios de manejo al o los administradores.

5.2. Recomendaciones

- El sistema debe ser periódicamente revisado debido a la necesidad de efectuar actualizaciones puesto que en el ámbito contable se cuenta con actualizaciones y cambios periódicos para poder cumplir

todos los nuevos requerimientos o modificar la funcionalidad del mismo de acuerdo sea el caso.

- Se debe brindar un proceso de capacitación tanto a los usuarios como al administrador para el correcto manejo del sistema, haciendo un énfasis en la persona que vaya a cumplir las funciones de administrador puesto que el tendrá el manejo completo del sistema.
- Para un sistema como el que se desarrolló, el apoyo de la empresa para la que se lo hizo es fundamental, puesto que al ser necesaria una explicación sobre varios temas concernientes al ámbito contable, de no contar con el apoyo suficiente el sistema no tendrá una finalización adecuada.

REFERENCIAS

[1] A. I. Hernádez y H. C. Pérez, Sistema de Calificación y Seguimiento de Pacientes del Centro ACASULCO.

[2] A. B. O. Varas, Desarrollo de un Aplicación Web utilizando la tecnología JSF con una arquitectura MVC para La Viandé.

[3] S. B. Mayoral, Desarrollo de un Aplicación Web en HTML5 para smartphones con los datos de bibliotecas públicas proporcionados por Opendatanavarra.

[4] M. Pilgrim, HTML5 Up and Running.

[5] M. D. N. y. c. individuales, «Secciones y Líneas Generales de un documento HTML5,» 27 Septiembre 2013. [En línea].

[6] J. Galloway, B. Wilson, S. Allen y D. Matson, Professional ASP.NET MVC5.

[7] R. Ruiz Limón, Historia y Evolución del Pensamiento Científico.

[8] F. Parot y R. Doron, Diccionario Akal de Psicología, Madrid.

[9] A. Romeo y L. Domenech, Materiales de Lengua y Literatura.

[10] D. Minguez Sanz y E. García Morales, Metodologías para el Desarrollo de Aplicaciones web: UWE.

[11] L. M. U. Munchen, «UWE - UML - based Web Engineering,» [En línea].

ANEXOS

Anexo 1 Manual de Usuario

Acceso al Sistema Principal (Administrador)

El administrador del sistema deberá ingresar directamente en la base de datos la información de la persona encargada de la parte administrativa del sistema, de la misma forma, proporcionará un nombre de usuario y contraseña para el ingreso a los diferentes módulos de la aplicación.



Manual Sesión Usuario

Elaborado por: Luis Torres

Posteriormente, el usuario procederá a ingresar a la aplicación con dicho usuario y contraseña proporcionados accediendo de esta forma al menú principal.



Manual Menú Principal

Elaborado por: Luis Torres

Ingresar un Nuevo Cliente

En el menú principal, el administrador se dirige a la pestaña de "*Administración"* donde elegirá la opción "*Clientes"* y posteriormente a *"Ingreso de Datos"* para ingresar los datos del nuevo contribuyente como se muestra a continuación.



Manual Registro de Clientes

Elaborado por: Luis Torres

El usuario ingresa los datos correspondientes al cliente y procede a presionar el botón *"Registrar"*, la aplicación muestra un resumen de todos los usuarios ingresados en el cuál podemos observar que el cliente ha sido ingresado correctamente.



Manual Cliente Ingresado Correctamente

Elaborado por: Luis Torres

Ingresar un Nuevo Usuario

En el menú principal, el administrador se dirige a la pestaña de "*Administración"* donde elegirá la opción "*Usuarios"* y posteriormente a *"Ingreso de Datos"* para ingresar los datos del nuevo contribuyente como se muestra a continuación.



Manual Registro de Usuarios

Elaborado por: Luis Torres

El usuario ingresa los datos correspondientes al cliente y procede a presionar el botón *"Registrar"*, la aplicación muestra un resumen de todos los

usuarios ingresados en el cuál podemos observar que el cliente ha sido ingresado correctamente.

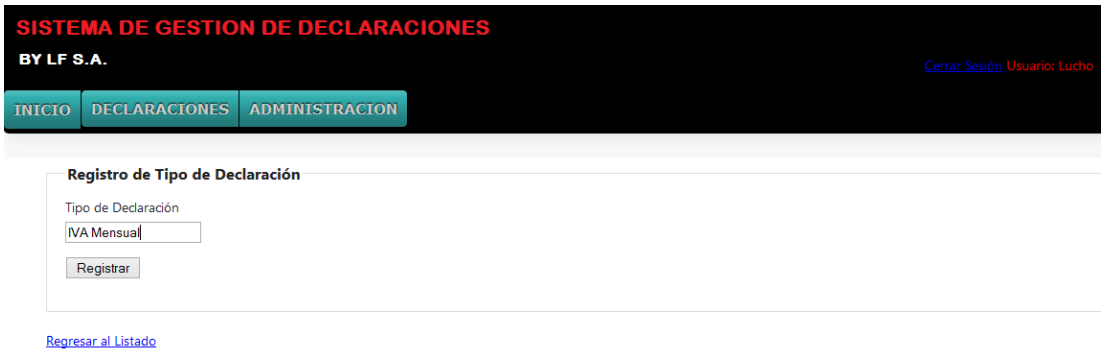


Manual Usuario Ingresado Correctamente

Elaborado por: Luis Torres

Ingresar Nuevo Tipo de Declaración

En el menú principal, el administrador se dirige a la pestaña de "*Administración"* donde elegirá la opción "*Tipos de Declaración"* y posteriormente a *"Ingreso de Datos"* para ingresar los datos del nuevo tipo de declaración como se muestra a continuación.

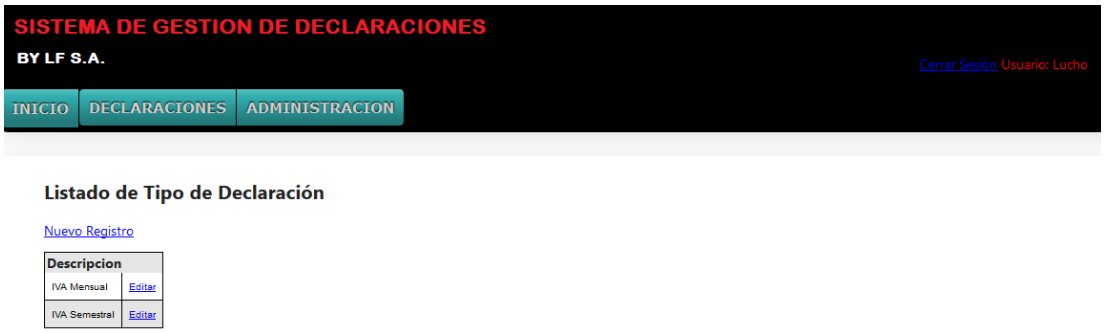


Manual Registro de Tipo de Declaración

Elaborado por: Luis Torres

El usuario ingresa los datos correspondientes al nuevo tipo de declaración y procede a presionar el botón *"Registrar"*, la aplicación muestra un resumen

de todos los tipos de declaración ingresados en el cuál podemos observar que el tipo de declaración ha sido ingresado correctamente.



Manual Tipo de Declaración Ingresado Correctamente

Elaborado por: Luis Torres

Ingresar Nueva Declaración

En el menú principal, el administrador se dirige a la pestaña de "*Declaraciones"* donde elegirá la opción "*Ingreso de Declaración"* para ingresar los datos del nuevo contribuyente como se muestra a continuación.

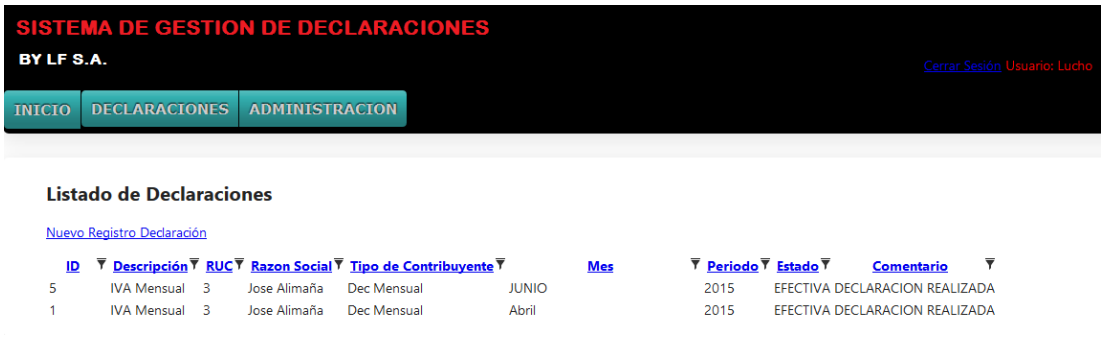


Manual Registro de Declaración

Elaborado por: Luis Torres

El usuario ingresa los datos correspondientes a la declaración y procede a presionar el botón *"Registrar"*, la aplicación muestra un resumen de todas las

declaraciones ingresadas en el cuál podemos observar que la declaración ha sido ingresada correctamente.



Manual Declaración Ingresada Correctamente

Elaborado por: Luis Torres

Exportar Datos

El usuario podrá obtener un resumen de todas las declaraciones realizadas, el mismo que podrá ser descargado para mantener un respaldo de dicha información, para esto el usuario se dirige a la pestaña de *"Declaraciones"*, donde elegirá la opción *"Exportar Datos"* y posteriormente a *"Datos Declaraciones"* como se muestra a continuación.



Manual Exportar Datos

Elaborado por: Luis Torres

Se presenta un resumen de las declaraciones elaboradas, para obtener el archivo de respaldo el usuario procede a presionar la opción "*Descargar Archivo"*, posteriormente se mostrará un cuadro de diálogo donde se da la

opción de guardar el archivo o simplemente verlo usando la herramienta predeterminada.



Manual Descarga Archivo Respaldo

Elaborado por: Luis Torre

Anexo 2 Código Fuente

Controlador Inicio

```
//Sección de código que se genera automáticamente al iniciar el proyecto,
//el modelo MVC provee código automático mientras se va avanzando en el mismo
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Conta;
using System.Web.Security;

namespace PQR_Web.Controllers
{
public class InicioController : Controller
{
private ContaEntities db = new ContaEntities();
//
// GET: /Inicio/
public ActionResult Index()
{
return View();
}
public ActionResult Login()
{
return View();
}
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Login(Usuario e)
{
```

```
if (ModelState.IsValid) //
{
using (ContaEntities db = new ContaEntities())
{
var v = db.Usuario.Where(p => p.RUC.Equals(e.RUC) &&
p.Contrasena.Equals(e.Contrasena)).FirstOrDefault();
if (v != null)
{
if (v.Estado == "ACTIVO")
{
FormsAuthentication.SetAuthCookie(v.Razon_Social, true);
@ViewBag.Perfil = Perfil (e.RUC.ToString());
return RedirectToAction("Index");
}
else
{
ViewBag.Message = "Usuario Inactivo...";
}
}
else
{
ViewBag.Message = "Datos incorrectos para inicio de Sesión...";
}
}
}
return View(e);
}
public ActionResult LogOut()
{
FormsAuthentication.SignOut();
return RedirectToAction("Index", "Inicio");
}
```

```
public String Perfil(string usuario)
{
String perfil = "";
var emp = db.Usuario.FirstOrDefault(e => e.RUC == usuario);
perfil = emp.Perfil;
return perfil;
}
}
}
```

Login Partial

```
@if (Request.IsAuthenticated) {
<text>
@Html.ActionLink("Cerrar Sesión", "LogOut", "Inicio")
Usuario: @User.Identity.Name
</text>
} else {
@Html.ActionLink("Inicio de Sesión", "Login", "Inicio")
}
```

Menú

Layout

```
@using PQR_Web;
@using PQR_Data;
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width" />
<meta name="description" content="" />
<meta name="keywords" content="" />
<title>@ViewBag.Title</title>
```

```
@* Llamado a librerias y procesos necesarios dentro del lenguaje *@
@Styles.Render("~/Content/css")
@Styles.Render("~/bundles/bootstrapcss")
@Scripts.Render("~/bundles/modernizr")
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/jqueryui")
@RenderSection("styles", required: false)
</head>
<body>
<div id="wrapper_enc">
<div id="header">
<div id="logo">
</div>
<div id="slogan">
<section id="login">
@Html.Partial("_LoginPartial")
</section>
</div>
</div>
@if (Request.IsAuthenticated)
{
<!-- Start css3menu.com BODY section -->
<input type="checkbox" id="css3menu-switcher" class="c3m-switch-input">
<ul id="css3menu1" class="topmenu">
<li class="switch"><label onclick="" for="css3menu-switcher"></label></li>
<li class="topfirst">@Html.ActionLink("INICIO", "Index", "Inicio")</li>
<li class="topfirst"><a href="#" title="Administración de Quejas y Reclamos" style="height:18px;line-height:18px;"><span>QUEJAS Y RECLAMOS</span></a>
<ul>
@using (PQREntities db = new PQREntities())
{
Funciones f = new Funciones();
```

```
if (f.Perfil(User.Identity.Name) == "ADMINISTRADOR")
{
<li class="subfirst">@Html.ActionLink("INGRESO DE DECLARACION", "Create", "Qr")</li>
}
}
<li>@Html.ActionLink("LISTADO GENERAL", "Index", "Qr")</li>
<li>@Html.ActionLink("SEGUIMIENTO", "Index", "Seguimiento")</li>
@using (PQREntities db = new PQREntities())
{
Funciones f = new Funciones();
if (f.Perfil(User.Identity.Name) == "ADMINISTRADOR")
{
<li><a href="#"><span>EXPORTAR DATOS</span></a>
<ul>
<li class="subfirst">@Html.ActionLink("DATOS QR", "Exportar", "Qr")</li>
<li>@Html.ActionLink("DATOS BITACORA QR", "ExportarBitacora", "Qr")</li>
<li>@Html.ActionLink("DATOS LOG QR", "", "")</li>
<li>@Html.ActionLink("DATOS INDECOPI", "ExportarTXT", "Qr")</li>
</ul></li>               }
}
</ul></li>
@using (PQREntities db = new PQREntities())
{
Funciones f = new Funciones();
if (f.Perfil(User.Identity.Name) == "ADMINISTRADOR")
{
<li class="toplast"><a href="#" title="Administración del Sitio" style="height:18px;line-
height:18px;"><span>ADMINISTRACION</span></a>
<ul>
<li><a href="#"><span>TIPOS DE DECLARACION</span></a>
<ul>
<li class="subfirst">@Html.ActionLink("INGRESO DE DATOS", "Create", "Canal")</li>
```

```
<li>@Html.ActionLink("LISTADO", "Index", "Canal")</li>
</ul></li>
<li><a href="#"><span>CLIENTES</span></a>
<ul>
<li class="subfirst">@Html.ActionLink("INGRESO DE DATOS", "Create", "Cliente")</li>
<li>@Html.ActionLink("LISTADO", "Index", "Cliente")</li>
</ul></li>
<li><a href="#"><span>USUARIOS</span></a>
<ul>
<li class="subfirst">@Html.ActionLink("INGRESO DE DATOS", "Create", "Empleado")</li>
<li>@Html.ActionLink("LISTADO", "Index", "Empleado")</li>
</ul></li>
</ul></li>
}
}
</ul>
<!-- End css3menu.com BODY section -->
}
</div>
<div id="wrapper">
<div id="page">
@RenderBody()
@Scripts.Render("~/bundles/jquery")
@RenderSection("scripts", required: false)
</div>
</div>
</body>
</html>
```

Modelos

Modelo Tipo de Declaración

```
//Código generado automáticamente en el que se llama las librerías necesarias para la aplicación

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using Conta;

using System.ComponentModel.DataAnnotations;

namespace PQR_Web.Models

{

public class Canal

{

public string ID { get; set; }

[Required(ErrorMessage = "* Campo Requerido")]

[Display(Name = "Tipo de Declaración")]

public string Descripcion { get; set; }

public virtual ICollection<Declaraciones> Declaraciones { get; set; }

}

}
```

Modelo Cliente

```
//Código generado automáticamente en el que se llama las librerías necesarias para la aplicación

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using Conta;

using System.ComponentModel.DataAnnotations;

namespace PQR_Web.Models
```

```
{
public class Cliente
{
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "RUC")]
public int RUC { get; set; }
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "Razón Social")]
public string Razon_Social { get; set; }
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "Correo Electrónico")]
public string Mail { get; set; }
public string Telf_Casa { get; set; }
public string Telf_Movil { get; set; }
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "Tipo de Contribuyente")]
public string Tipo_Contribuyente { get; set; }
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "Obligado a Llevar Contabilidad")]
public string Contabilidad { get; set; }
public virtual ICollection<Declaraciones> Declaraciones { get; set; }
}
}
```

Modelo Empleado

```
//Código generado automáticamente en el que se llama las librerías necesarias para la aplicación
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Conta;
using System.ComponentModel.DataAnnotations;
```

```
namespace PQR_Web.Models
{
public class Empleado
{
public int ID { get; set; }
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "RUC")]
public string RUC { get; set; }
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "Razón Social")]
public string Razon_Social { get; set; }
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "Contraseña")]
public string Contrasena { get; set; }
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "Perfil")]
public string Perfil { get; set; }
[Required(ErrorMessage = "* Campo Requerido")]
[Display(Name = "Estado")]
public string Estado { get; set; }
}
}
```

Modelo Declaraciones

```
//Código generado automáticamente en el que se llama las librerías necesarias para la aplicación
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using PQR_Data;
using System.ComponentModel.DataAnnotations;
namespace PQR_Web.Models
```

```
{
public class Qr
{
public int ID { get; set; }
[Required(ErrorMessage = "Campo  requerido")]
[Display(Name = "RUC")]
public string RUC_Cliente { get; set; }
[Required(ErrorMessage = "Campo  requerido")]
[Display(Name = "Tipo de Declaración")]
public int ID_Tipo_Declaracion { get; set; }
[Required(ErrorMessage = "Campo  requerido")]
[Display(Name = "Periodo")]
public string Periodo { get; set; }
[Required(ErrorMessage = "Campo  requerido")]
[Display(Name = "Mes")]
public string Mes { get; set; }
[Required(ErrorMessage = "Campo  requerido")]
[Display(Name = "Valor Pagado")]
public decimal Valor_Pagado { get; set; }
[Required(ErrorMessage = "Campo  requerido")]
[DataType(DataType.Date)]
//Declaración de tipos de datos mediante palabras propias del lenguaje
[DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
public System.DateTime Fecha { get; set; }
public Nullable<decimal> Multa { get; set; }
public virtual Cliente Cliente { get; set; }
public virtual Canal Tipo_Declaracion { get; set; }
public virtual ICollection<Seguimiento> Seguimiento { get; set; }
}
}
```

Vistas

Vista Crear Tipo de Declaración

```
@model PQR_Web.Models.Canal

@{

ViewBag.Title = "Create";

}

@* Código contiene secciones de declaración así como código propio de la aplicación generado automáticamente*@

@using (Html.BeginForm()) {

@Html.ValidationSummary(true)

<fieldset>

<legend>Registro de Tipo de Declaración</legend>

<div class="editor-label">

@Html.LabelFor(model => model.Descripcion)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.Descripcion)

@Html.ValidationMessageFor(model => model.Descripcion)

</div>

<p>

<input type="submit" value="Registrar" />

</p>

</fieldset>

}

<div>

@Html.ActionLink("Regresar al Listado", "Index")

</div>

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}
```

Vista Crear Cliente

```
@model PQR_Web.Models.Cliente

@{

ViewBag.Title = "Create";

}

@* Código contiene secciones de declaración así como código propio de la aplicación generado automáticamente*@

@using (Html.BeginForm()) {

@Html.ValidationSummary(true)


<fieldset>

<legend>Registro de Clientes</legend>

<div class="editor-label">

@Html.LabelFor(model => model.RUC)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.RUC)

@Html.ValidationMessageFor(model => model.RUC)

</div>

<div class="editor-label">

@Html.LabelFor(model => model.Razon_Social)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.Razon_Social)

@Html.ValidationMessageFor(model => model.Razon_Social)

</div>

<div class="editor-label">

@Html.LabelFor(model => model.Mail)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.Mail)

@Html.ValidationMessageFor(model => model.Mail)
```

```
</div>
<div id="LClienteNombres" class="editor-label">
@Html.LabelFor(model => model.Telf_Casa)
</div>
<div class="editor-field">
@Html.EditorFor(model => model.Telf_Casa)
@Html.ValidationMessageFor(model => model.Telf_Casa)
</div>
<div id="LClienteApellidoPaterno" class="editor-label">
@Html.LabelFor(model => model.Telf_Movil)
</div>
<div class="editor-field">
@Html.EditorFor(model => model.Telf_Movil)
@Html.ValidationMessageFor(model => model.Telf_Movil)
</div>
<div class="editor-label">
@Html.LabelFor(model => model.Tipo_Contribuyente)
</div>
<div class="editor-field">
@Html.DropDownListFor(model => model.Tipo_Contribuyente, new
SelectList(@ViewBag.Tipo_Contribuyente, "value", "text"), "Escoger Perfil")
@Html.ValidationMessageFor(model => model.Tipo_Contribuyente)
</div>
<div class="editor-label">
@Html.LabelFor(model => model.Contabilidad)
</div>
<div class="editor-field">
@Html.DropDownListFor(model => model.Contabilidad, new
SelectList(@ViewBag.CONTABILIDAD, "value", "text"), "Escoger Perfil")
@Html.ValidationMessageFor(model => model.Contabilidad)
</div>
<p>
<input type="submit" value="Registrar" />
```

```
</p>
</fieldset>
}
<div>
@Html.ActionLink("Regresar al Listado", "Index")
</div>
@section Scripts {
@Scripts.Render("~/bundles/jqueryval")
@*<script type="text/javascript">
$("#ClienteRazonSocial").hide();
$("#LRazonSocial").hide();
$("#TipoPersonaID").change(function () {
// this will call when Country Dropdown select change
var tipo_persona = $("#TipoPersonaID").val();
if (tipo_persona == 1) {
$("#ClienteRazonSocial").show();
$("#LRazonSocial").show();
$("#ClienteNombres").hide();
$("#LClienteNombres").hide();
$("#ClienteApellidoPaterno").hide();
$("#LClienteApellidoPaterno").hide();
$("#ClienteApellidoMaterno").hide();
$("#LClienteApellidoMaterno").hide();
}
else {
$("#ClienteRazonSocial").hide();
$("#LRazonSocial").hide();
$("#ClienteNombres").show();
$("#LClienteNombres").show();
$("#ClienteApellidoPaterno").show();
$("#LClienteApellidoPaterno").show();
$("#ClienteApellidoMaterno").show();
```

```
$("#LClienteApellidoMaterno").show();
}
});
</script>*@
}
```

Vista Crear Usuario

```
@model PQR_Web.Models.Empleado
@{
ViewBag.Title = "Create";
}
@* Código contiene secciones de declaración así como código propio de la aplicación generado automáticamente*@
@using (Html.BeginForm()) {
@Html.ValidationSummary(true)
<fieldset>
<legend>Registro de Usuarios</legend>
<div class="editor-label">
@Html.LabelFor(model => model.RUC)
</div>
<div class="editor-field">
@Html.EditorFor(model => model.RUC)
@Html.ValidationMessageFor(model => model.RUC)
</div>

<div class="editor-label">
@Html.LabelFor(model => model.Razon_Social)
</div>
<div class="editor-field">
@Html.EditorFor(model => model.Razon_Social)
@Html.ValidationMessageFor(model => model.Razon_Social)
</div>
<div class="editor-label">
```

```
@Html.LabelFor(model => model.Contrasena)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.Contrasena)

@Html.ValidationMessageFor(model => model.Contrasena)

</div>

<div class="editor-label">

@Html.LabelFor(model => model.Perfil)

</div>

<div class="editor-field">

@Html.DropDownListFor(model => model.Perfil, new SelectList(@ViewBag.PERFIL,
"value", "text"), "Escoger Perfil")

@Html.ValidationMessageFor(model => model.Perfil)

</div>

<div class="editor-label">

@Html.LabelFor(model => model.Estado)

</div>

<div class="editor-field">

@Html.DropDownListFor(model => model.Estado, new SelectList(@ViewBag.ESTADO,
"value","text"),"Escoger Estado")

@Html.ValidationMessageFor(model => model.Estado)

</div>
```

Vista Crear Declaración

```
@model PQR_Web.Models.Qr

@{

ViewBag.Title = "Create";

}

@* Código contiene secciones de declaración así como código propio de la aplicación
generado automáticamente*@


@using (Html.BeginForm()) {

@Html.ValidationSummary(true)
```

```
<fieldset>
<legend>Registro de Declaraciones</legend>
@Html.HiddenFor(model => model.ID)
<table>
<tr>
<td>
<div class="editor-label">
FECHA DE REGISTRO
</div>
<div class="editor-field">
@Html.EditorFor(model => model.Fecha)
@Html.ValidationMessageFor(model => model.Fecha)
</div>
</td>
</tr>
</table>
<table>
<tr>
<td>
<div class="editor-label">
CLIENTE
</div>
<div class="editor-field">
@Html.DropDownListFor(model => model.RUC_Cliente, new
SelectList(@ViewBag.ClienteID, "value", "text"), "Seleccionar Cliente")
@Html.ValidationMessageFor(model => model.RUC_Cliente)
</div>
</td>
<td>
<div class="editor-label">
TIPO DE DECLARACION
</div>
```

```
<div class="editor-field">

@Html.DropDownListFor(model=>model.ID_Tipo_Declaracion, new
SelectList(@ViewBag.Tipo_Declaracion,"value","text"), "Seleccionar Declaracion")

@Html.ValidationMessageFor(model => model.ID_Tipo_Declaracion)

</div>

</td>

<td>

<div class="editor-label">

PERIODO

</div>

<div class="editor-field">

@Html.DropDownListFor(model => model.Periodo, new SelectList(@ViewBag.PERIODO,
"value", "text"), "Seleccionar Periodo")

@Html.ValidationMessageFor(model => model.Periodo)

</div>

</td>

<td>

<div class="editor-label">

MES

</div>

<div class="editor-field">

@Html.DropDownListFor(model => model.Mes, new SelectList(@ViewBag.MES, "value",
"text"), "Seleccionar Mes")

@Html.ValidationMessageFor(model => model.Mes)

</div>

</td>

<td>

<div class="editor-label">

VALOR PAGADO

</div>

<div class="editor-field">

@Html.EditorFor(model => model.Valor_Pagado)

@Html.ValidationMessageFor(model => model.Valor_Pagado)
```

```
</div>
</td>
<td>
<div class="editor-label">
MULTA
</div>
<div class="editor-field">
@Html.EditorFor(model => model.Multa)
@Html.ValidationMessageFor(model => model.Multa)
</div>
</td>
</tr>
</table>
<p>
<input type="submit" value="Registrar" />
</p>
</fieldset>
}
<div>
@Html.ActionLink("Regresar al Listado", "Index")
</div>
@section Scripts {
@Scripts.Render("~/bundles/jqueryval")
}
```

Vista Exportar Declaraciones

```
@model IEnumerable<Conta.Vis_Exportar_Dec>
@{
ViewBag.Title = "Exportar";
}
<p>
@Html.ActionLink("Descargar Archivo", "Descargar")
```

```
</p>
<div class="CSSTableGenerator">
<table>
<tr>
<th>
ID
</th>
<th>
DESCRIPCION
</th>
<th>
RUC
</th>
<th>
RAZON SOCIAL
</th>
<th>
TIPO DE CONTRIBUYENTE
</th>
<th>
MES
</th>
<th>
PERIODO
</th>
<th>
ESTADO
</th>
<th>
COMENTARIO
</th>
</tr>
```

```
@foreach (var item in Model) {
<tr>
<td>
@item.ID
</td>
<td>
@item.Descripcion
</td>
<td>
@item.RUC
</td>
<td>
@item.Razon_Social
</td>
<td>
@item.Tipo_de_Contribuyente
</td>
<td>
@item.Mes
</td>
<td>
@item.Periodo
</td>
<td>
@item.Estado
</td>
<td>
@item.Comentario
</td>
</tr>
}
</table>
```

```
</div>
```

Web Config

```
<!--La configuración web posee código generado automáticamente por el lenguaje y es parametrizado con los datos necesarios para la aplicación como son el usuario y la contraseña del correo electrónico del que se va a enviar los correos-->

<?xml version="1.0"?>

<configuration>

<configSections>

<sectionGroup name="system.web.webPages.razor"
type="System.Web.WebPages.Razor.Configuration.RazorWebSectionGroup,
System.Web.WebPages.Razor, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35">

<section name="host" type="System.Web.WebPages.Razor.Configuration.HostSection,
System.Web.WebPages.Razor, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false" />

<section name="pages"
type="System.Web.WebPages.Razor.Configuration.RazorPagesSection,
System.Web.WebPages.Razor, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false" />

</sectionGroup>

</configSections>

<system.web.webPages.razor>

<host factoryType="System.Web.Mvc.MvcWebRazorHostFactory, System.Web.Mvc,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<pages pageBaseType="System.Web.Mvc.WebViewPage">

<namespaces>

<add namespace="System.Web.Mvc" />

<add namespace="System.Web.Mvc.Ajax" />

<add namespace="System.Web.Mvc.Html" />

<add namespace="System.Web.Optimization"/>

<add namespace="System.Web.Routing" />

</namespaces>

</pages>

</system.web.webPages.razor>

<appSettings>

<add key="webpages:Enabled" value="false" />
```

```
</appSettings>
<system.web>
<httpHandlers>
<add path="*" verb="*" type="System.Web.HttpNotFoundHandler"/>
</httpHandlers>
<!--
Enabling request validation in view pages would cause validation to occur after the input has already been processed by the controller. By default MVC performs request validation before a controller processes the input.

To change this behavior apply the ValidateInputAttribute to a controller or action.

<pages
validateRequest="false"
pageParserFilterType="System.Web.Mvc.ViewTypeParserFilter, System.Web.Mvc, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"
pageBaseType="System.Web.Mvc.ViewPage, System.Web.Mvc, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"
userControlBaseType="System.Web.Mvc.ViewUserControl, System.Web.Mvc, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
<controls>
<add assembly="System.Web.Mvc, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" namespace="System.Web.Mvc" tagPrefix="mvc" />
</controls>
</pages>
</system.web>
<system.webServer>
<validation validateIntegratedModeConfiguration="false" />
<handlers>
<remove name="BlockViewHandler"/>
<add name="BlockViewHandler" path="*" verb="*" preCondition="integratedMode" type="System.Web.HttpNotFoundHandler" />
</handlers>
</system.webServer>
</configuration>
```