**Capítulo 5 Ilustración de la propuesta con un enfoque de caso de estudio comparativo**

**5.1 INTRODUCCIÓN**

En este capítulo se describirá la experiencia del autor de la tesis en el proceso llevado a cabo para ilustrar el caso de estudio, en el cual se compara la capa de presentación de la metodología web MoWebA con extensiones RIA, con respecto a la misma metodología sin extensiones. La comparativa entre los enfoques tomados, se enmarca contextualmente en el dominio de las aplicaciones web, precisamente con la obtención de los datos analíticos, en base a la implementación de un sistema de administración de personas o *Person Manager*.

En la actualidad existen diversos métodos empíricos para llevar a cabo validaciones formales sobre algún fenómeno en particular, entre los que se pueden citar a los experimentos, las encuestas y los casos de estudio. Es común en el campo de la ingeniería de software emplear a los casos de estudio como métodos de validación, debido a su flexibilidad y a la posibilidad de tener un mejor control sobre las variables de medición, a costas de un mayor esfuerzo en la interpretación de los resultados obtenidos.

Según Runeson [1], un caso de estudio es llevado a cabo para investigar una sola entidad o un fenómeno en su contexto de la vida real, en un espacio de tiempo específico. Típicamente el fenómeno puede ser difícil de distinguir de su ambiente y el investigador debe colectar información detallada sobre un proyecto durante un periodo sostenido de tiempo. Durante la realización del caso de estudio, una variedad de procedimientos de colección de datos y perspectivas de análisis deben aplicarse.

Sin embargo, no siempre es posible tener el rigor suficiente para emplear un caso de estudio y es allí, donde surge como una alternativa; la ilustración, que a pesar de no ser un método de validación formal, sirve para presentar a una audiencia como funciona un artefacto y como este puede resolver un *toy problem* en particular, que permitirá de esta forma, llegar a una conclusión intuitiva[<r2014>]

En este capítulo, se ilustrará el aporte hecho a MoWebA presentando las extensiones realizadas a la metodología, por medio de la resolución de un *toy problem* denominado *Person Manager* desde los enfoques mencionados anteriormente. Para brindar a la ilustración propuesta una mayor formalidad, se seguirán las guías de caso de estudio propuestas por Runeson

**5.2 DISEÑO DEL CASO DE ESTUDIO**

**5.2.1 Razón fundamental**

Esta ilustración se presenta con la idea de llevar a cabo un análisis crítico de las extensiones RIA llevadas a cabo a la metodología web MoWebA, que forman parte del trabajo de fin de carrera del autor. Esta propuesta de extensión se basa principalmente en proveer a MoWebA de características enriquecidas a nivel de la interfaz de usuario, que le permitirán mantenerse vigente con respecto a las nuevas tendencias de las aplicaciones web de hoy en día, que demandan una mayor interactividad y riqueza en las interfaces de usuario.

Estas características RIA implementadas a MoWebA, ofrecen al usuario final la capacidad de llevar a cabo validaciones locales de datos en un formulario, inserciones de fechas de una manera amigable, resaltar un campo o texto de relevancia con un mensaje personalizado; como así también contraer y expander información para una mejor administración del espacio en las páginas, que son características comunes en las aplicaciones web RIA. Teniendo en cuenta, las potenciales mejoras que pueden llevarse a cabo con las extensiones propuestas, el propósito de este capítulo es resaltar estas nuevas características por medio de la implementación de un sistema que refleje estas extensiones y a la vez que sirva para poder efectuar un análisis comparativo de los cambios realizados a MoWebA, con respecto a la propuesta de presentación original.

**5.2.2 Objetivos**

La meta principal de la presente ilustración, es obtener datos lo suficientemente reveladores; qué permitan intuir que, la propuesta de extensión a nivel de la capa de presentación para el lado del cliente, llevada a cabo a la metodología web MoWebA, ofrece cobertura a algunas de las diversas características que contemplan las RIA que han sido analizadas en el capítulo 2. Puntualmente, estas características abarcan a la lógica de negocios en el lado del cliente, específicamente a las validaciones locales de campos en un formulario y a las presentaciones enriquecidas que contempla a ciertos eventos en el lado del cliente, *widgets* interactivos y el paradigma de una sola página o single *page paradigm*. El objetivo es analizar estas características, por medio de la resolución de un *toy problem* denominado *Person Manager* ,que es una aplicación web que contiene en sus especificaciones funcionales, características de las RIAS y resulta lo suficientemente expresiva para ilustrar la propuesta de extensión. **52.3 El caso y las unidades de análisis**

El caso ilustrativo está basado en un sistema de administración de personas (*Person Manager*) en el contexto de dominio de aplicaciones web, que fue elegido entre varias otras opciones, debido a que su alcance puede representar las características RIA que han sido agregadas a la metodología MoWebA, de una manera clara y simple.



Figura 1 Ilustración del sistema *Person Manager* implementado con MoWebA desde dos enfoques distintos

El caso será analizado desde dos unidades de análisis, en la cual la primera de ellas (el *enfoque A*); será implementar la capa de presentación del caso, con MoWebA sin extensiones RIAS y la otra (*el enfoque B*) será implementar la misma capa de presentación del caso, con la nueva propuesta de extensión RIA a MoWebA. Por lo tanto se trata también de un caso de estudio embebido, ya que se cuenta con más de una unidad de análisis para un mismo caso, como se puede apreciar en la Figura1.

A continuación se presenta la descripción del sistema *Person Manager* de una manera general, esto es; las funcionalidades básicas que deben contemplar ambas unidades de análisis. Las unidades de análisis representan a dos metodos implementados con MoWebA (A y B) que servirán para obtener respuestas a las preguntas de investigación que se presentaran en la siguiente sección. , Este sistema está inspirado en el trabajo de [<sv2008>]

Sistema de administración de personas(Person Manager)

Person Manager es una agenda de contactos personales, que contiene funciones de creación, listado y borrado de registros correspondiente a personas. Cada una de estas funciones contiene las siguientes vistas:

**Agregar(Add):** Consiste en una vista utilizada para capturar la suficiente información acerca de una persona para posteriormente agregarlo a una base de datos. En la vista, la información detallada de una persona en ingresado en un formulario. Al presionar el botón enviar (submit button), los datos ingresados en el formulario se insertaran en una base de datos. En la vista Agregar tenemos a los campos de entrada de texto, nombre, apellido, fecha de nacimiento, país de origen, email, usuario, clave, confirmación de clave; también al campo selección (radio) genero (masculino o femenino), y la caja de verificación de datos correctos (checkbock).

**Listar(List):** Consiste de una vista en la cual se despliegan todos los datos correspondientes a las personas existentes en la base de datos en una tabla. La tabla contiene una columna por cada campo de información que ha sido completado por un usuario en la vista Agregar .

**Borrar(Remove):** Consiste en una vista para borrar a una persona de la base de datos. El id de la persono a borrar es ingresado en un cuadro de texto .Al presionar el botón eliminar, el registro de la persona con el id especificado es eliminada de la base de datos y por ende desaparece de la vista Listar .

Adicionalmente a la definición común del Person Manager para ambas unidades de análisis mencionadas anteriormente, se adicionan los siguientes requerimientos RIAS, a ser tenido en cuenta por el enfoque B. Las características RIAS deseables son las siguientes:

Para los campos de la vista Agregar se requiere:

* Para el campo fecha de nacimiento de la persona, se desea que el ingreso de la fecha sea ágil e interactiva y que no sea necesario escribirla manualmente.
* En campos de entrada de texto o en hipervínculos en donde se requiera información adicional complementaria, es necesario agregar mensajes interactivos que provean de tal información, Esto es necesario en los siguientes:
  + Nombre: al posicionar el puntero del mouse sobre el campo nombre se debe desplegar el mensaje complementario “Ingrese su nombre completo”
  + Apellido: al posicionar el puntero del mouse sobre el campo apellido se debe desplegar el mensaje complementario “Ingrese apellido completo” y
  + *Clave:* al posicionar el puntero del mouse sobre el campo nombre se debe desplegar el mensaje complementario “La clave debe tener al menos 8 caracteres”
* En el campo de entrada de texto país de origen, a medida que caracteres son introducidos, sugerencias deben desplegarse en base al patrón actual, permitiendo al usuario, navegar en tales sugerencias, hasta elegir la opción deseada.
* Las siguientes validaciones locales (lado del cliente) de campos en un formulario pueden llevarse a cabo:
  + Validaciones de campos mandatorios: Para que el formulario pueda ser validado, los campos de entrada de texto nombre, apellido, contraseña, y confirmar contraseña deben ser completados de manera obligatoria, para el *radio choice* género debe seleccionarse una opción de las existentes (masculino, femenino ) y el *checkbox* que debe seleccionarse para la confirmación de acuerdo sobre los datos introducidos.
* Longitud mínima de caracteres en los siguientes campo de entrada:
  + Usuario: la longitud mínima debe ser de dos caracteres.
  + Contraseña: la longitud mínima debe ser de 8 caracteres.
* La validación de un email en el formato correcto para el campo email de la persona.
* Deben coincidir los dos campos de entrada del tipo contraseña, para la contraseña y la confirmación de contraseña.

Para el campo de la vista Eliminar se requiere:

* Para el campo id, se permite solamente el ingreso de números naturales (dígitos) (0,1,….,9, ..).
* Longitud máxima de caracteres en el campo de entrada id de la persona es de 10.
* El campo Id, es mandatorio.

La interfaz de usuario en forma general, debe ofrecer un aspecto *single page*, por lo tanto la navegación por las distintas vistas de la aplicación, no debe implicar un refrescado total de página. Todas las validaciones sobre los campos de entrada de los formularios que forman parte del *Person Manager* deben llevarse a cabo de manera local(en el lado del cliente). En caso de errores cometidos por no completar un campo (campo mandatorio), el/los mensaje/s de error, serán desplegados al lado del campo faltante, una vez presionado el botón *submit* del formulario en cuestión. En caso de cometer algún error en el ingreso de los datos, un mensaje de error en línea (al lado del campo en cuestión) se desplegará al desenfocar el campo, indicando al usuario el motivo del error, sin la necesidad de presionar el botón *submit*.

**5.2.4 Preguntas de investigación**

De los objetivos anteriormente citados, surgieron las siguientes preguntas de investigación para esta ilustración:

PI1: ¿Consume una mayor cantidad de tiempo emplear el método B que el método A?

PI2: Para la implementación del método B. ¿Es necesaria una mayor cantidad de generaciones de código para la obtención de la interfaz de usuario final, con respecto al método A?

PI3: ¿Qué ventajas aportan las características RIAS presentes en la aplicación implementada con el método B con respecto al método A, desde el punto de vista de las presentaciones enriquecidas?

PI4: Para cada una de las vistas del *Person Manager* ¿Que porciones de la interfaz de usuario han sido posibles de generar de manera automática a partir de los modelos, en los método A y B ?

PI5: ¿ Que tan independiente de la plataforma destino son los PIM presentados en el método B con respecto al método A?

**5.2.5 Conceptos**

Las variables de medición se definen a continuación:

Tiempo de modelado total empleado por el método A(*TMA*): El tiempo total en minutos, empleado para modelar la interfaz de usuario del Person Manager con MoWebA para el método A

Tiempo de modelado total empleado por el método B(*TMB*): El tiempo total en minutos, empleado para modelar la interfaz de usuario del Person Manager con MoWebA para el método B

Numero de generaciones de código para el método A (NGA): El número de veces que el código fuente de la aplicación *Person Manager* fue generado hasta obtener la interfaz de usuario final para el método A.

Numero de generaciones de código para el método B (NGB): El número de veces que el código fuente de la aplicación *Person Manager* fue generado hasta obtener la interfaz de usuario final para el método B.

**5.2.6 Métodos de colección de datos**

Este caso de estudio con enfoque ilustrativo, se implementó con una población de un individuo (el autor de la tesis) tanto para la implementación de las diferentes unidades de análisis, como en la colección de los datos), por lo tanto no se llevaron a cabo entrevistas ni encuestas para la colección de los datos de análisis.

Teniendo en cuenta este hecho particular, los datos necesarios para calcular las variables de respuesta, fueron colectados a medida que las unidades funcionales de la aplicación (las vistas) del *Person Manager*eran implementadas. Primeramente se colectó toda la información correspondiente al método A y luego se procedió a la colección de los datos del método B. Para cada uno de los métodos, primeramente se colectaron todos los datos correspondientes a una vista en particular hasta la conclusión de esta. Seguidamente se pasaba a la siguiente vista y se recaban los datos correspondientes y así sucesivamente.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nombre de las vistas | NGA | NGB | TMA | Tiempo (horas) con extesnsiones | Tiempo sin extensiones | TMB | Errores cometidos extensiones |
| Agregar persona |  |  |  | Inicio: 5:42pm  Fin:  6:38pm | Inicio:5:57am  Fin: 6:14  am |  |  |
| Mostrar persona |  |  |  | Inicio 1340  Fin  1408 | Inicio: 5:25am  Fin: 5:47am |  |  |
| Remover persona |  |  |  | Inico 1415  Fin 1530 | Inicio:4:55am  Fin:  5:14am |  |  |
| Totales |  |  |  |  |  |  |  |

Una vez finalizada la implementación de ambas unidades de análisis del caso, de manera adicional se llevó a cabo un análisis de líneas de código, para medir el tamaño de los proyectos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Generadas | Correctivos | Perfectivos | Adaptativos | Total LDC |
| Manualmente  Automáticamente |  |  |  |  |
| Total LDC |  |  |  |  |

**5.2.7 Métodos de análisis de los datos**

En vista que el método de comparación es entre proyectos (croos-proyect) o proyecto hermano (*sister project*), solo podemos tener una variable para la productividad o calidad por tratamiento. En este caso no se pueden utilizar métodos estadísticos para asegurar si las diferencias entre las variables de respuesta son significativas. Por lo tanto, solo se pueden comparar los valores obtenidos del proyecto de control informalmente con los valores obtenidos del tratamiento hecho al proyecto, según las recomendaciones hechas en Klitchenan.

Se llevará a cabo análisis cuantitativos de los datos obtenidos.

**5.2.8 Selección del caso y minimización de los factores de confusión**

Como se mencionó anteriormente, se llevó a cabo una ilustración de caso de estudio comparativo, en la cual se optó por un proyecto piloto en el contexto de las aplicaciones web , la cual fué implementado por un mismo equipo de trabajo (el autor). Se optó por el proyecto piloto *Person Manager*en la cual es posible representar las nuevas características extendidas a MoWebA de manera clara y concisa y teniendo en cuenta que dentro de los métodos de comparación existentes en la actualidad, según Klitchenan, el cross-project o single-project, es el más incurre en costos( en el caso de esta propuesta el mayor costo es el tiempo de desarrollo), ya que un mismo proyecto debe llevarse a cabo por un mismo equipo (el autor en este caso) con más de un enfoque o unidad de análisis (en esta ilustración, se utilizaron dos unidades de análisis) . Con respecto a los factores tenidos en cuenta para minimizar los factores de confusión, se ha tenido en consideración los siguientes puntos:

1. Se ha llevado a cabo pruebas de modelado y generación de código para cada uno de los elementos que forman parte del perfil de contenido de MoWebA para lograr la mayor familiaridad posible con cada uno de los elementos de interfaz y sus propiedades intrínsecas. Esto fue necesario (a pesar de que el autor haya sido el implementador de las extensiones RIAS en los perfiles de modelado y las plantillas de transformación ) para minimizar la cantidad de errores de modelado en cada método implementado y de paso testear la plantilla de transformación y reducir los riesgos de posibles *bugs* que podrían darse en el transcurso de las pruebas llevadas a cabo en el proceso de desarrollo de cada método.
2. Primeramente se empleó completamente el método A hasta obtener la interfaz final del *Person Manager* sin extensiones RIA y luego el método B con extensiones RIA de igual manera. Se tomó esta decisión, ya que si se implementa una vista con el método A y luego la misma vista con el método B, se puede obtener demasiado familiaridad con el modelado de la vista en cuestión, y por ende no sería tan realista la medición de los tiempos de modelado en el método B, ya que estos podrían reducirse.
3. Los métodos A y B fueron utilizados para construir el Person Manager con una semana diferencia, implementándolas hasta obtener la interfaz final desde cero.
4. Cuando un método estaba en proceso, se iba implementando cada una de las vistas y a la par se iba recabando los datos. Se pasaba a implementar la vista siguiente de la aplicación, una vez culminada en su totalidad la vista actual con todos los datos analíticos recabados.

**5.2.10 Selección de los datos**

Los datos analíticos fueron obtenidos a medida que el desarrollo que los proyectos a ser comparados se ibandesarrollando, para posteriormente, una vez seleccionados de las planillas correspondientes, se pueda proceder al análisis y de esa forma concluir los resultados obtenidos.

**5.2.11 Definición y almacenamiento de los Datos**

Todos los datos recabados en el desarrollo de cada uno de los métodos,fueron almacenados y mantenidos en las plantillas creadas para el efecto..

**5.2.13 Mantenimiento del protocolo de caso de estudio.**

Este caso de estudio no se rige por un protocolo formal de evaluación.

**5.2.14 Reporte y difusión de los datos obtenidos en la ilustración del caso de estudio.**

**En esta sección se responderán a las preguntas de investigación presentadas en la sección 5.2.4 que se citan a continuación:**

**PI3: ¿Qué ventajas aportan las características RIAS presentes en la aplicación implementada con el método B con respecto al método A, desde el punto de vista de las presentaciones enriquecidas?**

Son diversos los aportes ventajosos con respecto al método A, que pueden distinguirse en la interfaz RIA obtenida por medio del método B. A continuación se presentan algunas ventajas.

***Apariencia de una aplicación single page***

Cada una de las páginas que forman parte de la aplicación *Person Manager* implementada con el método B, son equivalentes a las pestañas pertenecientes a un *widget richTabs*. Por lo tanto cuando se navega en la aplicación, se tiene la sensación de que trata de una aplicación de escritorio, ya que se puede recorrer cada una de las pestañas sin necesidad de un refrescado de página, teniendo toda la información de manera local en una sola página. Esta característica mejora la interactividad con el usuario de la aplicación y el look and feel del mismo. En la implementación llevada a cabo con el método A, cada una de las páginas de la aplicación, está representada por un enlace, y por ende, cada vez que se visita una página de la aplicación, un refrescado total de página se lleva a cabo, perdiéndose de esta forma el concepto de *single page*.

Estas mismas ventajas de manera similar al RichTab, también pueden ser aprovechadas al utilizar la extensión *RichAccordion*

***Optimización del espacio y navegabilidad de las páginas***

Con la extensión *RichTab,* es posible encapsular distintos elementos de interfaz presentes en MoWebA, tales como entradas de texto, enlaces, botones, textos, hipervínculos, formularios y tablas, como así también elementos enriquecidos que son parte de la extensión; como los *RichDatePicker*, *RichToolTips*, *RichAutocompletes*. La extensión de validación de campos *RichFieldLiveValidation*, también puede ser utilizada dentro de un RichTab, por ejemplo para la validación de los campos en un formulario. La posibilidad de encapsular muchos elementos de interfaz dentro de cada una de las pestañas que forman parte de un *RichTab*, resulta ventajoso con respecto a la optimización espacial de los elementos dentro de las páginas.

Estas mismas ventajas de manera similar al RichTab, también pueden ser aprovechadas al utilizar la extensión *RichAccordion*

***Validaciones locales de los diversos campos de un formulario.***

La ventaja principal de llevar a cabo validaciones en los formularios de manera local, es que no es necesario ninguna interacción con el lado servidor, lo cual mejora el rendimiento de la aplicación, evitando retardos al recargar la página tras la solicitud de envío de los datos. Con los *richFieldLiveValidation* es posible llevar a cabo validaciones a los diversos campos de los formularios de la aplicación *PersonManager*. Dentro de las validaciones que se han efectuado se muestra primeramente la validación en los campos que son mandatorios y que no pueden quedar vacios como puede apreciarse en la figura

Seguidamente se efectuaron controles locales sobre la cantidad de caracteres que deben tener como mínimo algunos campos, tales como los de usuario, password y confirmación de password, que se han configurado como mínimo en 8 caracteres. En contraparte para los campos nombre y apellido se controló que estos no excedan una cantidad máxima de 30 caracteres. Para los campos password y confirmación de password, también se verificó que ambos coincidan en los valores introducidos.

Para los campos numéricos, se valida que solamente sea posible el ingreso de dígitos (valores del 0 al 9), por ejemplo; en el campo id, utilizado para borrar un registro del sistema. En este campo de igual manera no es posible ingresar más de 12 dígitos para evitar algún desbordamiento numérico en la base de datos. También, el campo email verifica que la cadena ingresada por el usuario corresponda a un email válido.

Por último, en el campo de selección; como el de género, es mandatorio seleccionar uno de los radio controles (masculino, femenino), como así también, es mandatorio seleccionar la caja de selección del campo de conformidad. Los datos introducidos en el formulario solo serán enviados al servidor, cuando todos los campos pasen la validación correspondiente a cada uno de ellos.

***Widgets interactivos en la interfaz de usuario***

***Datepicker***

El datepicker permite desplegar de una manera interactiva un calendario debajo de la entrada textual que corresponde al campo fecha de nacimiento. Este calendario interactivo ofrece la posibilidad de navegar por los distintos meses del año actual, con las flechas indicadoras izquierda y derecha, como así también, permite seleccionar un mes en particular desplegando una lista de meses. Con respecto a los años del calendario, es posible definir un rango de años que podrán seleccionarse de igual manera de una lista desplegable.

***Autocomplete***

El autocomplete ofrece la posibilidad de desplegar un listado de opciones que facilitan al usuario la escritura de texto en un cuadro de texto de entrada. En el campo país de origen, a medida que el usuario va introduciendo caracteres correspondientes al país deseado, interactivamente se despliegan todas las opciones que coinciden con el patrón introducido, permitiendo navegar de arriba abajo por medio de un cursor sobre las distintas opciones. El cursor se resalta con un color diferente a medida que se va recorriendo por las sugerencias. Una vez que el usuario encuentra el país deseado de origen, al presionar la tecla entrar o al hacer clic sobre la opción, esta es escrita en el cuadro de texto.

***ToolTip***

Con el toolTip un mensaje informativo útil al usuario es desplegado al posar el puntero del mouse sobre un texto de entrada en particular. Para los campos nombre y apellido se muestra un mensaje en el que se indica al usuario que se ingrese el nombre y el apellido completo. Para el caso del campo contraseña, se despliega al usuario a modo de sugerencia, el mensaje de seguridad que indica el ingreso de caracteres alfanuméricos con mayúsculas y minúsculas combinados con caracteres especiales y que contenga por lo menos una longitud de ocho caracteres.

Este paso está pendiente

**5.3 ASUNTOS LEGALES, ÉTICOS Y PROFESIONALES**

Puesto que la ilustración de caso de estudio es parte de un proyecto de tesis, todos los datos obtenidos quedan regidos bajo las normas expuestas por la universidad. Teniendo en cuenta que una sola persona implementó las unidades de análisis y a la vez recabó los datos analíticos, era fundamental llevar adelante cada paso, con la mayor transparencia y objetividad posible, para que los resultados obtenidos sean fidedignos y de valor. Es bajo esta circunstancia, que decidió llevar a cabo una ilustración y no un caso de estudio, ya que los resultados y conclusiones obtenidas, se dejan a la intuición y criterio de la audiencia, y no cuentan con el rigor que conlleva un caso de estudio.

**5.4 CONCLUSION**