**Capítulo 5**

**Ilustración DE LA PROPUESTA**

**5.1 INTRODUCCIÓN**

En este capítulo se describirá el proceso llevado a cabo para realizar una validación preliminar de MoWebA con extensiones RIA. La validación consiste en comparar 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*.

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[<c2012>].

Según Runeson [<p2012>], 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.

Atendiendo a la anterior definición, no siempre es posible realizar un caso de estudio. Una alternativa que se puede utilizar es la ilustración, que a pesar de no ser un método de validación formal, sirve para presentar a una audiencia cómo funciona un artefacto y cómo este puede resolver un *toy problem* en particular, permitiendo llegar a una conclusión intuitiva [<r2014>].

En este capítulo se utiliza una ilustración para validar preliminarmente las extensiones RIA de MoWebA por medio de la resolución de un *toy problem* denominado *Person Manager*. Para brindar mayor formalidad a la ilustración, la misma se realizó siguiendo las guías propuestas por Runeson para realizar casos de estudio.

**5.2 DISEÑO DE LA ILUSTRACIÓN**

**5.2.1 Objetivos**

Esta ilustración se presenta con la idea de ofrecer un análisis crítico de las extensiones RIA llevadas a cabo a la metodología Web MoWebA desde el punto de vista de las interfaces enriquecidas. . La 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 expandir 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 ofrecer 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 que a la vez sirva para poder efectuar un análisis comparativo de los cambios realizados a MoWebA con RIA con respecto a MoWebA sin RIA.

La ilustración busca obtener datos lo suficientemente reveladores que 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 contemplan 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*. El *Person Manager* es una aplicación Web que contiene en sus especificaciones funcionales características de las RIA y resulta lo suficientemente expresiva para ilustrar la propuesta de extensión.

**5.2.1.1 Objetivos específicos**

Comparar a MoWebA sin RIA y a MoWebA con extensiones RIA, con respecto al tiempo de modelado y a la cantidad de generaciones y refinamientos a nivel de modelado que se deben hacer hasta obtener una interfaz final satisfactoria.

Verificar si MoWebA con extensiones RIA, ofrece ventajas sobre MoWebA sin RIA con respecto a las presentaciones enriquecidas y con respecto a la lógica en el lado del cliente.

Analizar la cantidad de líneas de código generadas de manera automática a partir de los modelos PIM, para ambos enfoques.

**5.2.4 Preguntas de investigación**

A partir de los objetivos anteriormente citados, surgen las siguientes preguntas de investigación para esta ilustración:

PI1: ¿Consume una mayor cantidad de tiempo modelar la aplicación aplicando MoWebA con RIA que MoWebA sin RIA?

PI2: Para cual de los enfoques es necesaria una mayor cantidad de generaciones de código para la obtención de la interfaz de usuario final?

PI3: Desde el punto de vista de las presentaciones enriquecidas, ¿qué ventajas aportan las características RIA presentes en la aplicación implementada con MoWebA con RIA con respecto a MoWebA sin RIA?

PI4: Desde el punto de vista de la lógica de negocios en el lado del cliente, ¿qué ventajas aportan las características RIA presentes en la aplicación implementada con MoWeba con RIA con respecto a MoWebA sin RIA?

PI5: Para cada una de las vistas del *Person Manager,* ¿qué cantidad de líneas de código para la interfaz de usuario se pudieron generar de manera automática a partir de los modelos, para ambos enfoques?

**5.2.2 El caso y las unidades de análisis**

El caso ilustrativo está basado en un sistema de administración de personas (*Person Manager*) en el dominio de las aplicaciones Web, que fue elegido entre varias otras opciones debido a que sus requerimientos funcionales ofrecen la posibilidad de representar a todas las características RIA que han sido agregadas a la metodología MoWebA, de una manera clara y sencilla.



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

El caso fue analizado desde dos unidades de análisis. La primera unidad de análisis se refiere a la implementación de la capa de presentación del *Person Manager* con MoWebA sin RIA. La segunda unidad de análisis se refiere a la implementación de la misma capa de presentación del caso estudiado, con la nueva propuesta de extensión RIA a MoWebA. Por lo tanto, se trata de una ilustración embebida, ya que se cuenta con más de una unidad de análisis para un mismo caso, como se puede apreciar en la Figura 1.

En el Anexo 1 se presenta la descripción del sistema *Person Manager* de una manera general, es decir, se describen los requerimientos funcionales básicos y se agregan al *Person Manager* algunos requerimientos adicionales, que son precisamente, requerimientos RIA. Las unidades de análisis representan a los dos métodos implementados con MoWebA con o sin RIA que servirán para obtener respuestas a las preguntas de investigación . El *Person Manager* está basado en el trabajo de Gharavi [<sv2008>].

**5.2.5 Conceptos**

Las variables de medición necesarias para responder las primeras dos preguntas de investigación se definen a continuación:

*TMA*: El tiempo total en minutos empleado para modelar la interfaz de usuario del *Person Manager* aplicando el método A. Corresponde a la suma de todos los tiempos de modelado de cada una de las vistas implementadas con el método A.

*TMB*: El tiempo total en minutos empleado para modelar la interfaz de usuario del *Person Manager* aplicando el método B. Corresponde a la suma de todos los tiempos de modelado de cada una de las vistas implementadas con el método B.

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.

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.

Una generación de código (equivalente a compilar la aplicación) es llevada a cabo para obtener la interfaz final a partir de los modelos de entrada. En cada una de ellas, el código fuente se va refinando a partir de la modificación de los modelos de entrada.

**5.2.6 Métodos de colección de datos**

Esta ilustración se realizó 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 recababan los datos correspondientes y así sucesivamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de las vistas** | **TMA** | **NGA** | **TMB** | **NGB** |
| **Agregar persona** |  |  |  |  |
| **Mostrar persona** |  |  |  |  |
| **Remover persona** |  |  |  |  |
| **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.

**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, el cual fue implementado por un mismo equipo de trabajo (el autor). Se optó por el proyecto piloto *Person Manager,* en el cual es posible representar las nuevas características extendidas a MoWebA de manera clara y concisa. 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 han 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 demasiada 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 iban 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.

Teniendo en cuenta que el autor del trabajo implementó las unidades de análisis y a la vez recabó los datos analíticos, se trató de llevar adelante cada paso con la mayor transparencia y objetividad posible, para que los resultados obtenidos sean fidedignos y de valor. Sin embargo esto no es suficiente para otorgar la suficiente formalidad a los resultados obtenidos. Es bajo esta circunstancia, que se 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.

**5.2.14 Análisis e interpretación de los resultados.**

En esta sección se responderán las preguntas de investigación presentadas en la sección 5.2.4.



PI1: ¿Consume una mayor cantidad de tiempo modelar la aplicación utilizando el método B que el método A?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de las vistas** | **TMA** | **NGA** | **TMB** | **NGB** |
| **Agregar persona** | 50 | 3 | 56 | 4 |
| **Mostrar persona** | 27 | 1 | 28 | 1 |
| **Remover persona** | 29 | 2 | 30 | 3 |
| **Totales** | 106 | 6 | 116 | 8 |

En la Tabla 1 pueden apreciarse los tiempos de modelado obtenidos usando ambas metodologías.

Tabla 1, Tiempos de modelado y numero de generaciones para cada una de las Vistas del *Person Manager*

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?

Se pudo apreciar que el método B deparó en una mayor cantidad de generaciones de código para obtener la interfaz RIA final. Esta diferencia representa un aumento del 20% con respecto a la implementación llevada a cabo con el método A. Analizando las generaciones de cada una de las vistas del *Person Manager* de la Tabla 2, se puede notar que la vista que tuvo que generarse una mayor cantidad de veces fue la vista Agregar Persona, y este dato resulta concordante con lo que puede intuirse preliminarmente, ya que esta vista es la que contiene la mayor cantidad de requerimientos de interfazy por ende existe una mayor probabilidad de cometer fallos en el modelado, lo que incurre en una mayor cantidad de veces que la aplicación debe generarse hasta su depuración final. Para la vista Remover Persona se incurrió en un número mínimamente superior de generaciones de código implementando el método B, precisamente un 10% más que con el método A, pudiendo deberse también, a que la vista con el método B contiene requerimientos de interfaz RIA a diferencia del método A. También el número de generaciones disminuyó en ambos métodos con respecto a la vista Agregar Persona. En la vista Listar Personas, se tuvo la mínima cantidad de generaciones de código en ambos métodos aplicados, debido a que gran parte de ella es implementada de manera manual.

De los resultados presentados puede intuirse que a mayor requerimientos de interfaz, se requiere una mayor cantidad de generaciones de código para ambas metodologías aplicadas para ir depurando la aplicación, con un leve incremento en el caso de la metodología B y esto podría deberse a que dada una mayor cantidad de detalles a especificar en los modelos de entrada de la aplicación, existe una mayor posibilidad de cometer fallos.

**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 richTab*. 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 relacionadas al RichTab, también pueden ser aprovechadas al utilizar la extensión *RichAccordion.*

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

***Datepicker***

El *datepicker* permite desplegar de una manera ágil e 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.

Este *widget* resulta de gran ayuda a los usuarios finales, ya que gráfica e intuitivamente permite seleccionar una fecha, evitando a estos cometer errores innecesarios al digitar una fecha en un formato dado y optimizando su tiempo de interacción con las páginas Web.

***AutoSuggest***

El *autoSuggest* 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 todos los países que coinciden con el patrón introducido, permitiendo navegar de arriba a abajo por medio de un cursor sobre los distintos países. El cursor se resalta con un color diferente a medida que se va recorriendo por los países sugeridos. Una vez que el usuario encuentra el país de origen deseado, al presionar la tecla entrar o al hacer clic sobre el país, este se escribe en el cuadro de texto de entrada.

***ToolTip***

A menudo es útil complementar con información adicional los campos de entrada de los formularios Web. Con el toolTip, un mensaje informativo útil al usuario es desplegado al posar el puntero del mouse sobre un cuadro de texto de entrada en particular. Para el *Person Manager*, en 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 solicita el ingreso de caracteres alfanuméricos con mayúsculas y minúsculas combinados con caracteres especiales y que contenga por lo menos una longitud de ocho caracteres.

***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 RIA propuesta a MoWebA, como los *RichDatePicker*, *RichToolTip*, *RichAutoSuggest*. La extensión de validación de campos *RichFieldLiveValidation* también puede ser utilizada dentro de un RichTab, 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 relacionadas al RichTab, también pueden ser aprovechadas al utilizar la extensión *RichAccordion.*

***PI4: ¿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 la lógica de negocios en el lado del cliente?***

Cuando se habla de lógica de negocios en el lado del cliente, hablamos de operaciones complejas y específicas para un dominio en particular, como así también de validaciones sobre los datos de entrada. Las extensiones RIAS propuestas a MoWebA abarcan específicamente a las validaciones sobre los campos de entrada en los formularios.

***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 el elemento *richFieldLiveValidation* es posible llevar a cabo validaciones a los diversos campos de los formularios de la aplicación *Person Manager*. Dentro de las validaciones que se han efectuado se muestra primeramente la validación en los campos que son obligatorias mandatorios y que no pueden quedar vacios.

Seguidamente se efectuaron controles locales sobre la cantidad de caracteres que deben tener como mínimo algunos campos, tales como los de usuario, clave y confirmación de clave, que se han configurado como mínimo en 2 y en 8 caracteres respectivamente. En contraparte, para los campos nombre y apellido se verificó que estos no excedan una cantidad máxima de 30 caracteres. Para los campos clave y confirmación de clave, 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 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.

.

**5.4 RESUMEN DEL CAPITULO**

En este capítulo se presentó una ilustración en parte y en la medida de lo posible, siguiendo las guías propuestas para llevar a cabo un caso de estudio y para brindar cierta formalidad Para tal motivo se implementó una aplicación denominada *Person Manager* en la cual pueden reflejarse muchas de las extensiones RIA llevadas a cabo a la metodología Web MoWebA, que es el aporte principal de este trabajo de fin de carrera. El *Person Manager* fue implementado desde dos enfoques distintos, precisamente para modelar la presentación que abarca al Contenido y Estructura de las páginas. En uno de ellos se utilizó a la metodología MoWebA en su forma original. En el otro enfoque se utilizó a MoWebAcon las extensiones RIA.

El *Person Manager* fue separado en tres vistas que fueron analizadas y en la cual se recabaron datos analíticos para responder a preguntas de investigación. Esos datos correspondían