



FACULTAD DE INGENIERÍAS Y CIENCIAS AGROPECUARIAS

SISTEMA DE COTIZACIÓN DE UNIFORMES MEDIANTE CATÁLOGO EN
LÍNEA EMPLEANDO SILVERLIGHT (WPF/E) Y XAML.

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Sistemas de Computación
e Informática

Profesor guía:
Santiago Albuja

Autor:
Juan David Utreras Jácome

Año
2009

Declaración del Profesor Guía

Yo, Ing. Santiago Albuja, con cédula de identidad No. 1710245711 certifico haber dirigido el trabajo de titulación del alumno Juan David Utreras Jácome, bajo el título de “Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”.

Agradecimiento

A Dios y a mi familia por estar siempre a mi lado apoyándome y animándome; sin importar la situación geográfica.

A Silvia Ascazubi, que con su cariño supo darme ánimo para seguir.

A mi profesor y amigo Ing. Santiago Albuja; quien me guió e impulsó durante todo este proceso, apoyándome con su conocimiento y aún con los recursos tecnológicos para el desarrollo de este proyecto.

A mi amigo Ing. Rodolfo Capelo; que me dio las pautas para la construcción del aplicativo del proyecto.

A mis profesores, Ing. Juan Chamorro, Ing. Juan Carlos Trujillo; Ing. Mario Valarezo; Ing. Juan José León y a Carlos Galeas; quienes me transmitieron los conocimientos que han determinado mi visión profesional. Y en forma general a todos y cada uno de mis maestros quienes se han constituido en un gran referente a seguir.

Finalmente, y sin restar importancia, a mis amigos que soportaron mis atrasos, faltas y excusas durante todo este tiempo de tesis; animándome en cada momento y apoyándome siempre con su amistad y oraciones.

Gracias.

Dedicatoria

A mi único y soberano Dios, Jesucristo; a quien debo todo lo que tengo y soy. A mis padres David Utreras y Olga Jácome; quienes nunca dejaron de creer en mí, y gracias a su esfuerzo sobre humano y su apoyo inmerecido este sueño se ha cumplido. Y por último a mis hermanos Josué y Francis; que son a más de mi apoyo; mi reto continuó a ser mejor.

Resumen

El presente proyecto titulado “Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”, es la implementación de un producto de software orientado a la Web; que promociona los productos de la empresa Donaire Needle Point.; el cual proporciona la funcionalidad de generación inmediata de cotizaciones para uniformes mediante un subsistema que permite la personalización de prendas por parte del cliente.

En lo referente al análisis y diseño del sistema se empleó la metodología RUP, y UML. Contemplando el uso de artefactos que subdividen al sistema en módulos para una mejor concepción funcional.

Considerando la ventaja de que la metodología es iterativa se inició la etapa de concepción del sistema fundamentando su existencia en la documentación proporcionada por la empresa y requerimientos de usuario. Se definió posteriormente el uso de herramientas como “Visual Web Developer ES” en conjunto las suites, Microsoft Expressions y Adobe CS3. Para las etapas de elaboración y construcción; se consideró entornos de desarrollo multicapa que incluyeron trabajo en conjunto con bases de datos como Microsoft SQL. Todo esto haciendo un énfasis especial en el proceso investigativo de la tecnología Silverlight; y efectuado una demostración práctica de las herramientas antes mencionadas.

Índice

DECLARACIÓN DEL PROFESOR GUÍA	I
AGRADECIMIENTO	II
DEDICATORIA	III
RESUMEN	IV
ÍNDICE	V
CAPÍTULO I	1
1.1. INTRODUCCIÓN.....	1
1.2. JUSTIFICACIÓN DEL PROYECTO.....	3
1.3. OBJETIVOS DEL PROYECTO.....	4
CAPÍTULO II	6
MARCO TEÓRICO	6
2.1 WINDOWS PRESENTATION FOUNDATION.....	6
2.1.1 CARACTERÍSTICAS DE WPF.....	7
2.1.2 Tecnología Windows Presentation Foundation.....	8
2.1.3 Aplicación de Windows Presentation Foundation.....	18
2.2 SILVERLIGHT.....	25
2.2.1 Definición y Origen de Silverlight.....	26
2.2.2 Estructura de Silverlight.....	28
2.2.3 Arquitectura de Silverlight.....	31
2.2.4 Propiedades y Componentes de Silverlight.....	34
2.2.5 Herramientas para Silverlight.....	36
2.2.6 Requerimientos de Software para ejecución de contenido Silverlight.....	39
2.2.7 Lenguajes que interactúan con Silverlight.....	40
2.2.7.1. JavaScript.....	40
2.2.7.2. ASP.NET.....	44
2.3 XAML.....	57
2.3.1 Funcionamiento.....	58
2.3.2 Sintaxis.....	61

2.3.3	<i>Extensiones de marcado específicas de WPF</i>	65
2.3.4	<i>Procesadores de XAML</i>	67
2.4	AJAX	67
2.4.1	<i>Funcionamiento</i>	69
2.4.2	<i>Navegadores que soportan AJAX</i>	72
2.5	UML	74
2.5.1	<i>Importancia del Modelamiento</i>	76
2.5.2	<i>Diagramas de UML</i>	78
2.5.2.1.	Diagrama de casos de uso	78
2.5.2.2.	Diagrama de Clases.....	81
2.5.2.3.	Diagrama de Componentes	82
2.5.2.4.	Diagrama de Despliegue	84
2.5.2.5.	Diagramas de Secuencia.....	85
2.5.2.6.	Diagramas de Colaboración	87
2.5.2.7.	Diagramas de Actividad	88
2.5.2.8.	Diagramas de Estados	90
2.6	RUP	92
2.6.1	<i>Características</i>	93
2.6.2	<i>Principios de desarrollo</i>	94
2.6.3	<i>Ciclo de vida</i>	95
2.6.4	<i>Elementos de RUP</i>	100
2.6.5	<i>Artefactos de RUP</i>	101
2.6.6	<i>Modelos</i>	104
CAPÍTULO III		106
ANÁLISIS DEL SISTEMA		106
3.1	DESCRIPCIÓN DEL PROBLEMA	106
3.2	DOCUMENTO DE VISIÓN	108
3.2.1	<i>Introducción</i>	111
3.2.2	<i>Propósito</i>	111
3.2.3	<i>Alcance</i>	111

3.2.4	<i>Posicionamiento del Producto</i>	113
3.2.5	<i>Descripciones de Afectados y Usuarios</i>	117
3.2.6	<i>Resumen del Producto</i>	119
3.3	RESUMEN EJECUTIVO TÉCNICO DEL SISTEMA.....	123
3.4	ESPECIFICACIONES SUPLEMENTARIAS.....	126
3.5	ESPECIFICACIONES DE CASOS DE USO	126
3.5.1	<i>Breve Descripción</i>	126
3.5.2	<i>Precondiciones</i>	126
3.5.3	<i>Pos condiciones de éxito</i>	127
3.5.4	<i>Pos condiciones de falla</i>	127
3.5.5	<i>Actor Principal</i>	128
3.5.6	<i>Flujo de Eventos</i>	128
3.5.6.1.	<i>Flujo Básico (FB)</i>	128
3.5.7	<i>Diagramas</i>	130
3.5.8	<i>Relaciones</i>	132
3.6	DIAGRAMAS DE CASOS DE USO	133
3.6.1	<i>Diagrama de casos de uso General</i>	134
3.6.2	<i>Diagrama de casos de uso por módulos</i>	134
3.7	DIAGRAMAS DE SECUENCIA PRINCIPALES	136
3.7.1	<i>Gestionar Cotizaciones</i>	137
3.8	DIAGRAMAS DE SECUENCIA SECUNDARIOS	140
3.9	DIAGRAMAS DE COLABORACIÓN DE ANÁLISIS	140
3.10	DIAGRAMAS DE COLABORACIÓN DE DISEÑO	140
3.11	DIAGRAMAS DE ESTADO	140
3.12	DIAGRAMAS DE CLASES	141
3.13	MODELO DE DATOS.....	144
	<i>Modelo Físico</i>	147
3.14	IMPLEMENTACIÓN DE CÓDIGO.....	148
3.15	DIAGRAMA DE COMPONENTE	148
3.16	DIAGRAMA DE DESPLIEGUE.....	148

3.17	RESUMEN DE CASOS DE PRUEBAS	148
3.18	GLOSARIO DE TÉRMINOS	148
3.19	ANÁLISIS DE COSTOS DEL PROYECTO	149
CAPÍTULO IV		158
CONCLUSIONES		158
CAPÍTULO V		160
RECOMENDACIONES		160
CAPÍTULO VI		162
BIBLIOGRAFIA		162
6.1	RECURSOS WEB	162
6.2	RECURSOS TEXTUALES	162
CAPÍTULO VII.....		1
ANEXOS		1
7.1.	ANEXO “A”: ESPECIFICACIONES SUPLEMENTARIAS	2
7.1.1	<i>Propósito</i>	3
7.1.2	<i>Alcance</i>	3
7.1.3	<i>Vista General</i>	3
7.1.4	<i>Funcionalidad</i>	3
7.1.4.1	<i>Consultar Catálogo</i>	3
7.1.4.2	<i>Cargar el contenido según la categoría.</i>	4
7.1.4.3	<i>Registrar Clientes.</i>	4
7.1.4.4	<i>Crear contraseñas de usuario para el acceso</i>	4
7.1.5	<i>Usabilidad</i>	5
7.1.6	<i>Desempeño</i>	7
7.1.7	<i>Soporte</i>	7
7.1.8	<i>Restricciones de diseño</i>	8
7.1.9	<i>Interfaces</i>	8
7.2.	ANEXO “B”: ESPECIFICACIONES DE CASOS DE USO	9
7.2.1	<i>Registrar Usuario</i>	10

7.2.2	<i>Consultar Catálogo</i>	13
7.2.3	<i>Validar Usuario</i>	18
7.2.4	<i>Gestionar Imágenes</i>	21
7.2.5	<i>Gestionar Categorías</i>	26
7.2.6	<i>Gestionar Productos</i>	35
7.2.7	<i>Gestionar Novedades</i>	43
7.3.	ANEXO “C”: DIAGRAMAS DE SECUENCIA SECUNDARIOS	48
7.3.1	<i>Registrar Usuario</i>	49
7.3.2	<i>Consultar Catálogo</i>	52
7.3.3	<i>Validar Usuario</i>	54
7.3.4	<i>Gestionar Imágenes</i>	55
7.3.5	<i>Gestionar Categorías</i>	58
7.3.6	<i>Gestionar Productos</i>	61
7.3.7	<i>Gestionar Novedades</i>	63
7.4.	ANEXO “D”: DIAGRAMAS DE COLABORACIÓN DE ANÁLISIS	65
7.4.1	<i>Gestionar Cotizaciones</i>	66
7.4.2	<i>Registrar Usuario</i>	69
7.4.3	<i>Consultar Catálogo</i>	70
7.4.4	<i>Validar Usuario</i>	71
7.4.5	<i>Gestionar Imágenes</i>	72
7.4.6	<i>Gestionar Categorías</i>	72
7.4.7	<i>Gestionar Productos</i>	73
7.4.8	<i>Gestionar Novedades</i>	74
7.5.	ANEXO “E”: DIAGRAMAS DE COLABORACIÓN DE DISEÑO	75
7.5.1	<i>Gestionar Cotizaciones</i>	76
7.5.2	<i>Registrar Usuario</i>	79
7.5.3	<i>Consultar Catálogo</i>	80
7.5.4	<i>Validar Usuario</i>	81
7.5.5	<i>Gestionar Imágenes</i>	81
7.5.6	<i>Gestionar Categorías</i>	82

	X
7.5.7	<i>Gestionar Productos</i> 83
7.5.8	<i>Gestionar Novedades</i> 84
7.6.	ANEXO “F”: DIAGRAMAS DE ESTADO 85
7.6.1	<i>Gestionar Cotizaciones</i> 86
7.6.2	<i>Registrar Usuario</i> 86
7.6.3	<i>Consultar Catálogo</i> 87
7.6.4	<i>Validar Usuario</i> 87
7.6.5	<i>Gestionar Imágenes</i> 87
7.6.6	<i>Gestionar Categorías</i> 88
7.6.7	<i>Gestionar Productos</i> 88
7.6.8	<i>Gestionar Novedades</i> 89
7.7.	ANEXO “G”: IMPLEMENTACIÓN DE CÓDIGO 90
7.8.1	<i>Implementación De Código</i> 91
7.8.2	<i>FASE 1 – ASP.NET, Flash</i> 92
7.8.3	<i>FASE 2 - ASP.NET, Silverlight</i> 94
7.8.4	<i>FASE 3 – Store Procedure SQL (MSQL Express)</i> 99
7.8.	ANEXO “H”: DIAGRAMA DE COMPONENTES..... 101
	<i>Diagrama De Componentes</i> 102
7.9.	ANEXO “I”: DIAGRAMA DE DESPLIEGUE 103
	<i>Diagrama De Despliegue</i> 104
7.10.	ANEXO “J”: RESUMEN DE CASOS DE PRUEBAS 104
7.10.1	<i>Caso de Uso: Gestionar Cotizaciones</i> 105
7.10.2	<i>Caso de Uso: Registrar Usuario</i> 111
7.10.3	<i>Caso de Uso: Consultar Catálogo</i> 115
7.10.4	<i>Caso de Uso: Validar Usuario</i> 119
7.10.5	<i>Caso de Uso: Gestionar Categorías</i> 122
7.10.6	<i>Caso de Uso: Gestionar Imágenes</i> 126
7.10.7	<i>Caso de Uso: Gestionar Productos</i> 130
7.10.8	<i>Caso de Uso: Gestionar Novedades</i> 134
7.11.	ANEXO “K”: GLOSARIO DE TÉRMINOS 138

CAPÍTULO I

Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML

1.1. Introducción

El presente proyecto pretende cubrir la necesidad de automatizar procesos mediante un sistema de catálogo en línea para la empresa Donaire Needle Point. Como tal, este sistema contempla el uso de algunas de algunas de las últimas tecnologías y herramientas que la Corporación Microsoft ha lanzado al mercado para el desarrollo de aplicaciones Web, en conjunto con otras tecnologías actualmente disponibles para el desarrollo de este tipo de aplicaciones.

Una de las tecnologías empleadas en el presente proyecto, es Silverlight o WPF/E. Esta propuesta orientada a Interfaces de usuario, está compuesta por lenguajes de programación como XAML, Java Script y C#; los cuales son compatibles con ASP.NET y HTML.

Para las etapas de diseño y análisis del sistema; se contempla el uso de la Metodología RUP y el lenguaje unificado de Modelamiento (UML); recursos que permiten una concepción general del sistema, separándolo en módulos de acuerdo a cada categoría, enfocando de una manera versátil la funcionalidad

del mismo al plantear un modelo que subdivide: Catálogo detallado de Productos, Administración de Catálogo, y Manejo de cotizaciones. Cada uno de estos módulos contiene los elementos necesarios referentes a Interfaces de Usuario.

El sistema de catálogo como tal solucionará el problema de promoción de productos de la empresa; y brindará además la posibilidad de generar cotizaciones de uniformes según parámetros definidos por el cliente.

El catálogo constará de un subsistema de cotización, el cual permitirá al usuario modificar según su conveniencia modelos de prendas predefinidos, agregando o quitando elementos que compongan el uniforme hasta conseguir una idea global del producto requerido.

Para esto se considera el uso de la tecnología AJAX, que permite potenciar en gran manera el uso de eventos como “Post Back” dentro de los formularios de ASP.NET los mismos que ejecutan operaciones en tiempo real. Permittiéndole al usuario obtener cotizaciones en ese mismo instante; tal y como el sistema lo demostrará en su implementación.

1.2. Justificación Del Proyecto

La implementación de aplicaciones empleando tecnologías como Silverlight (WPF/E), ASP.NET, AJAX, entre otras; constituye un gran desafío profesional ya que, por el nivel de desarrollo tecnológico en nuestros días y los efectos de la globalización nuevos recursos son lanzados al mercado diariamente, exigiendo una permanente actualización profesional y sin que deje de existir el riesgo de que pretendiendo aplicar una herramienta que se considera innovadora, en el proceso de desarrollo de software, aparezcan alternativas o en su defecto obstáculos que exijan reorientar los proyectos con la finalidad de obtener mejores resultados.

La empresa Donaire Needle Point, requiere la implementación de un sistema de información, automático e integrado; que centralice y consolide los datos referentes a: clientes, precios de productos, calidades de materiales y modelos de prendas entre otros.

Esto se debe a que el éxito o fracaso de cualquier empresa en la actualidad; depende en gran medida de los sistemas informáticos que esta emplee para realizar sus procesos tanto de producción como de comercialización.

Si estos sistemas no se encuentran totalmente automatizados e integrados de manera funcional; el rendimiento y productividad de la empresa serán deficientes.

Por tanto, el presente proyecto de investigación, pretende satisfacer esta necesidad mediante una aplicación Web tipo catálogo que a más de promocionar las líneas de productos en sus diversas categorías y calidades; permita la generación de cotizaciones aproximadas de acuerdo a parámetros escogidos por el cliente, agilitando de esta forma el proceso de promoción y obtención de cotizaciones para quien requiere uno o varios productos en las categorías que la empresa ofrece.

1.3. Objetivos del Proyecto

Objetivo General

Analizar, diseñar e implementar un sistema de catálogo en línea para la empresa Donaire Needle Point; el mismo que promocioe sus productos y permita la generación de cotizaciones según los parámetros escogidos por el usuario; empleando la tecnología Silverlight, en conjunto con ASP.NET, AJAX; entre otras.

Objetivos Específicos

- Verificar la información existente en la empresa Donaire Needle Point en lo referente a las áreas administrativas y de comercialización, con el fin de permitir el desarrollo del proyecto.

- Implementar una interface de usuario, práctica y amigable para el sistema de catálogo en línea; que promocione los productos y servicios de la empresa. Aplicando la tecnología; Silverlight (WPF/E), y XAML.
- Implementar en la aplicación de catálogo en línea, un sistema de cotización de uniformes; que genere proformas referenciales según parámetros escogidos por el cliente en base a elementos predefinidos.
- Emplear la metodología RUP para el desarrollo del Sistema.

CAPÍTULO II

MARCO TEÓRICO

Como respaldo a la presente investigación, se efectúa una revisión teórica de las herramientas que se utilizan en el desarrollo de los distintos módulos del proyecto.

2.1 Windows Presentation Foundation

Windows Presentation Foundation (WPF), conocida formalmente como AVALON; Provee un modelo de programación unificado que permite crear aplicaciones con interfaces ricas en contenido; las mismas que pueden ser integradas para su funcionamiento en conjunto con aplicaciones robustas y seguras como las que propone el entorno .NET y la incorporación de documentos, componentes multimedia, gráficos bidimensionales y tridimensionales, animaciones, características tipo Web, etc. Como esta descrito para su difusión:

“WPF consigue la combinación de los mejores atributos de sistemas como DirectX (plano tridimensional y aceleración de hardware), Windows Forms (Productividad de Diseño), Adobe Flash (Soporte para animación) y HTML (fácil desarrollo de “declarative markup”) haciéndola una tecnología robusta. Pese a esto es necesario mencionar que WPF aun está en etapa de pruebas y desarrollo, por tanto es una herramienta nueva que debe solucionar ciertos problemas y limitaciones.”¹

WPF es una solución para desarrolladores de sistemas y diseñadores gráficos que deseen crear interfaces de usuario sin tener la necesidad de entrar en

¹ Adam Nathan y Daniel Lehenbauer. Windows Presentation Foundation Unleashed”.

conflictos de complejidad tecnológica; gracias a que está compuesta por un motor interno que acopla los recursos de software y hardware empleando un conjunto de clases administradas que los desarrolladores pueden usar para crear aplicaciones.

2.1.1 CARACTERISTICAS DE WPF

Funcionalidad.- Tiene un amplio nivel de integración porque está fundamentada en el lenguaje XAML; su independencia de resolución permite el tratamiento de imágenes sin afectar su definición debido a que está compuesto por elementos vectoriales.

Aceleración de Hardware.- Permite que las gráficas bidimensionales, tridimensionales o tipo texto; gráficos de tipo hardware, se ejecuten de manera eficiente e independiente, porque posee la capacidad de convertir los elementos a texturas triangulares tridimensionales, que facultan una mejor ejecución de gráficas sin afectar el rendimiento del hardware empleado.

Programación Declarativa.- Se emplea para la implementación de la tecnología el lenguaje XAML.

“Extensible Application Markup Language”²

² Petzold Charles. C# 2008 for Programmers: Harvey M. Deitel, Edit. Prentice Hall,2008

Amplio nivel de Optimización y Composición de Controles y Componentes.- Los controles desarrollados para WPF y Silverlight permiten la exposición de imágenes y videos en tiempo real.

WPF potencia además el uso de “Windows Forms” dentro del entorno .NET, implementando nuevas funcionalidades que integran el uso y desarrollo de recursos como los descritos en la tabla 1.1 la que se presenta a continuación.

“Plataforma Unificada de Windows Presentation Foundation”

	Windows Forms	PDF	Windows Forms / GDI+	Windows Media Player	Direct3D	WPF
Interface gráfica, como formularios y controles	X	X
Documentos en pantalla	X	X
Documentos de formato fijo	.	X	.	.	.	X
Imágenes	.	.	.	X	.	X
Vídeo y audio	.	.	.	X	.	X
Gráficos bidimensionales	.	.	X	.	.	X
Gráficos tridimensionales	X	X

Tabla 1.1

2.1.2 Tecnología Windows Presentation Foundation

Modelo de aplicación

EL Framework de .NET como modelo de aplicación es estandarizado; y como tal organiza toda su funcionalidad en un grupo de espacios de nombres, System.Windows.

La clase Aplicación estándar de WPF; es la raíz de la cual se heredan todos los servicios comunes y útiles para cualquier aplicación, incluyendo estados de bloqueo y métodos estándar como RUN con todas sus funcionalidades. Los objetos Application se pueden crear con XAML.

Las aplicaciones WPF utilizan la misma clase raíz, y permiten escoger dos modalidades para el desarrollo de interfaces; una tradicional basada en cuadros de diálogo y una de navegación para una aplicación; brindando el mismo nivel de ventajas y opciones independientemente del estilo de interface.

Diseño y Controles.- WPF emplea paneles de diseño. Cada panel puede contener elementos secundarios, como controles (por ejemplo, botones y cuadros de texto) y otros paneles. Estos pueden clasificarse principalmente en "DockPanel", "Grid" y "Canvas"

Entre los controles estándar se incluyen "Button", "Label", "TextBox", "ListBox", "Menu", Slider y otros componentes tradicionales. Existen además controles más complejos como "SpellCheck", "PasswordBox" y otros para el trabajo con dispositivos como "Tablet PC". Además se puede crear controles personalizados partiendo de los controles base ya existentes.

Como es usual en interfaces gráficas, los controles incluidos en las aplicaciones de WPF pueden detectar y controlar eventos generados por el usuario (por ejemplo, los movimientos del ratón y el uso de teclas). Mientras

que los controles y otros elementos de interface de usuario pueden especificarse por completo mediante XAML, los eventos requieren el uso de código.

WPF maneja su estructura mediante el uso de árboles creados mediante la combinación de XAML y código. Los elementos que componen el árbol se almacenan como objetos de modo que WPF cuenta con la información suficiente para controlar este tipo de representación, incluso en casos de modificación del tamaño de ventanas o de los controles contenidos, WPF dispone de la información necesaria para volver a crear la interface según el nuevo tamaño, ya que es capaz de interpretar la forma de los gráficos (líneas, elipses, etc.) y se basa en gráficos vectoriales, no en mapas de píxeles.

Respecto a Estilos y plantillas, WPF emplea un recurso similar a los estilos en cascada (CSS) empleados en HTML, recurso que permite la creación de una apariencia uniforme para la interface. “Style” es el elemento de XAML, que permite definir uno o varios aspectos de la apariencia de un elemento particular y posteriormente aplicar el mismo estilo una y otra vez. WPF también admite el uso de plantillas. Las plantillas son similares a los estilos y se subdividen en dos tipos disponibles:

- **Plantillas de datos.-** permiten el uso del elemento “DataTemplate” de XAML para especificar conjuntos de características relacionadas con la apariencia de datos que incluyen color, alineación entre otros.
- **Plantillas de control.-** Permiten el uso del elemento “ControlTemplate” de XAML para definir la apariencia de controles.

Texto.- WPF ofrece compatibilidad con “OpenType”, lo que permite el uso de las bibliotecas de fuentes existentes; y es compatible con la tecnología “ClearType”, que suaviza la apariencia del texto mediante la colocación de sub píxeles, una técnica usada en la iluminación de los subelementos individuales que forman parte de cada píxel. Además, WPF proporciona compatibilidad de bajo nivel para la representación de texto mediante la clase “Glyphs”.

Para mejorar aún más la legibilidad, WPF se sirve elementos adicionales como las ligaduras, que hacen posible la sustitución de grupos de caracteres por imágenes conectadas únicas.

Documentos.- WPF subdivide la representación de documentos de la siguiente manera:

Documentos Fijos.- Presentan la misma apariencia tanto en pantalla como en formato impreso. El contenido de los documentos fijos se puede especificar mediante el elemento “FixedDocument” de XAML. Este sencillo

elemento contiene una lista de elementos “PageContent,” cada uno de los cuales incluye el nombre de una página del documento fijo.

Para mostrar un documento fijo, WPF cuenta con el control “DocumentViewer,” que ofrece una visualización de sólo lectura de documentos XPS para permitir al lector desplazarse por el documento, realizar búsquedas de fragmentos de texto determinados, etc.

Documentos de Flujo.- La presentación de los documentos de flujo está ideada únicamente para la pantalla. Con el fin de que su contenido resulte lo más claro posible, la apariencia del texto y de los gráficos se puede ajustar automáticamente al tamaño de la ventana, entre otros factores. Los documentos de flujo vienen definidos por el elemento “FlowDocument” de XAML el mismo que en WPF proporciona varios controles; que son: FlowDocumentReader, FlowDocumentPageViewer, y FlowDocumentScrollViewer

Imágenes.- Los archivos pueden incluir información sobre las mismas (metadatos), como palabras clave y clasificaciones definidas por los usuarios. El control “Image” en WPF, permite leer y escribir esta información. Incluso es posible incorporar estas en objetos tridimensionales. Además este control admite imágenes almacenadas en varios formatos, entre ellos, JPEG, BMP,

TIFF, GIF y PNG. Incluyendo también el formato “Windows Media Photo” (WMPhoto).

Vídeo y audio.- WPF ofrece compatibilidad integrada para estos dos elementos mediante el uso del control “MediaElement”; el cual permite reproducir vídeo en formato WMV, MPEG y AVI, junto con varios formatos de audio, mediante CODECS específicos para cada tipo de archivo.

Gráficos bidimensionales.- En lo que respecta a estos gráficos WPF define un grupo de formas que las aplicaciones pueden usar para crear imágenes. Todos estos recursos visuales están orientados directamente a la composición de piezas gráficas vectoriales.

Entre las principales herramientas para dibujo de formas en WPF están:

Line: dibuja una línea recta entre dos puntos.

Ellipse: dibuja una elipse.

Rectangle: dibuja un rectángulo.

Polygon: dibuja una forma cerrada definida por un grupo de líneas rectas conectadas entre sí.

Polyline: dibuja una forma abierta definida por un grupo de líneas rectas conectadas entre sí.

Path: dibuja formas descritas por una ruta arbitraria.

Existen otros pinceles, tales como: degradado vertical, degradado radial, pinceles que usan imágenes, mapas de bits, etc. Es necesario mencionar que las formas también admiten el uso de lápices para especificar el color, el ancho y el estilo del contorno.

En WPF, todo está creado sobre una base común, por lo que las aplicaciones pueden mostrar elementos “Image” dentro de elementos “Rectangle”, colocar elementos “Ellipse” dentro de elementos “Button”, etc. Esto incluye la combinación de imágenes de tipo bidimensional y tridimensional

Gráficos tridimensionales.- El trabajo en planos tridimensionales requiere consideraciones especiales como ubicación, desplazamiento, planos, texturas, manejo de sólidos, etc. Estos factores agregan un nivel de complejidad considerable al desarrollo de aplicaciones. WPF pretende cambiar este panorama al incluir la compatibilidad con gráficos 3D en el entorno estándar.

En el entorno de Windows se emplea el componente API de Direct 3D para el desarrollo de interfaces tridimensionales. WPF, emplea esta API de igual manera con la diferencia que lo hace de forma no visible en un entorno más sencillo. Para mostrar las gráficas tridimensionales WPF utiliza el control “Viewport3D”.

La creación de estas interfaces se realiza describiendo uno o más modelos y a continuación se asignan las propiedades de cada elemento como iluminación y visualización todo esto especificado mediante XAML.

WPF incluye la clase “GeometryModel3D” para la descripción de modelos, la cual permite definir la forma de los mismos. Una vez definido el modelo; su apariencia se controla mediante la aplicación de diferentes tipos de material.

Estas subclases son:

SpecularMaterial: dota de brillo a las superficies.

DiffuseMaterial: quita brillo a las superficies.

DirectionalLight: proporciona luz desde una dirección específica.

AmbientLight: permite la iluminación uniforme de todo lo que contiene una escena.

PerspectiveCamera: sirve para especificar la distancia y la perspectiva de visualización de los modelos.

OrthographicCamera: es similar a “Perspective Camera”, excepto en lo que concierne a la perspectiva: los objetos que están más alejados de la cámara no parecen más pequeños.

La creación de escenas tridimensionales complejas de forma directa mediante XAML o con código, no es un proceso sencillo. Por lo que en muchos de los casos se emplea software auxiliar específico que permita este desarrollo.

Transformación y efectos.- WPF permite la asignación de atributos especiales a los elementos visuales como giros, traslaciones y cambios en tamaño. En XAML, esto se consigue mediante los elementos “RotateTransform” y “ScaleTransform”, respectivamente. Este tipo de transformación se puede aplicar a cualquier elemento de interface de usuario. Asimismo, WPF incluye algunos efectos predefinidos.

Animaciones.- Puede ser muy útil dotar de movimiento a los elementos de la interface, es decir, convertirlos en elementos animados. Al igual que las transformaciones, las animaciones se pueden aplicar a muchos aspectos de interface de usuario: botones, formas, imágenes, etc.

Para crear una animación, basta con cambiar el valor de una o más propiedades de un objeto a lo largo de cierto período de tiempo mediante controles como TimeLine.

La definición de un grupo de animaciones relacionadas suele ser de utilidad. Para ello, WPF incluye la clase “Storyboard”. Cada elemento, puede incluir varias escalas de tiempo diferentes que, a su vez, pueden contener varias animaciones, lo que permite la ejecución de las animaciones en secuencia o en paralelo.

Enlace de datos.- WPF permite la conexión directa de la apariencia de los controles de con datos externos. Por ejemplo, el valor de la propiedad “Text” en un control “TextBox” de WPF se puede enlazar a la propiedad “Name” de un objeto “Employee”. Los cambios que se produzcan en una propiedad se podrán reflejar posteriormente en la otra. Es decir que, si se actualiza el valor de “TextBox”, la propiedad “Name” del objeto “Employee” también cambiará y viceversa. La creación de este tipo de conexión entre las propiedades de dos objetos requiere el uso de la clase “Binding” de WPF.

También existen opciones de enlace de datos más complejas como rellenar el contenido de controles “ListBox” a partir de cualquier objeto CLR que implemente la interface “IEnumerable” estándar.

De ser necesario, también se pueden filtrar u ordenar los datos antes de presentarlos. Aunque el enlace a un “Dataset” de ADO.NET es perfectamente posible, WPF no ofrece compatibilidad directa con el enlace de datos en sistemas de administración de bases de datos relacionales.

Independientemente de la opción de enlace de datos que se use, la intención es conseguir que la presentación de datos en interfaces de usuario siga un proceso lo más directo posible.

Para información más detallada respecto a la tecnología Windows Presentation Foundation se recomienda consultar sitios de Internet referentes al tema por ejemplo el sitio oficial de “MSDN”³

2.1.3 Aplicación de Windows Presentation Foundation

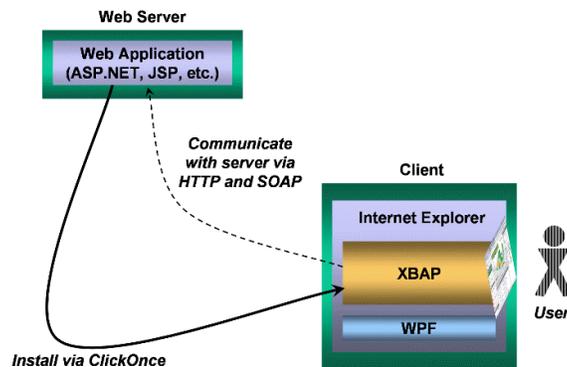
Actualmente, la tecnología WPF se aplica principalmente en las seis siguientes formas:

Aplicaciones de WPF independientes.- Se ejecutan como cualquier otra aplicación de Windows y no es necesario disponer de un explorador Web. Debido a esto no tienen limitantes en acceso a funciones, recursos específicos de sistema o incluso servicios disponibles en el equipo, como Windows Communication Foundation (WCF). Para el proceso de instalación de este tipo de aplicaciones se emplea la función “ClickOnce” de “.NET Framework”. Y al igual que otras aplicaciones de Windows; ésta se puede realizar desde un disco local o desde un servidor de red.

Aplicaciones XAML de explorador XBAP.- Aunque las aplicaciones de WPF independientes ofrecen la máxima funcionalidad, no siempre son la mejor elección. En muchas circunstancias, la ejecución de un cliente en un explorador Web resulta más útil que ejecutarlo como aplicación de Windows. WPF incluye aplicaciones XBAP para este tipo de aplicaciones.

³ Sitio Oficial MSDN – www.msdn.com

Como muestra la figura 2.1, las XBAP se ejecutan en Internet Explorer. Las XBAP pueden actuar como cliente en aplicaciones Web creadas con ASP.NET, “JavaServer Pages” (JSP) u otras tecnologías Web.



Ejecución de Aplicación XBAP en Internet Explorer

Figura 2.1

Estas aplicaciones pueden usar “HTTP” o SOAP para devolver la comunicación a la aplicación Web, independientemente de la plataforma de servidor que se use, la carga de las XBAP se realiza siempre mediante “ClickOnce”.

Durante este proceso, no se presentarán cuadros de diálogo ni mensajes al usuario; las XBAP se cargan de forma idéntica a las páginas Web, por este motivo, las XBAP no aparecen en el menú Inicio ni en el menú de “Agregar o quitar programas” del sistema operativo Windows.

Aunque no se trata de un requisito estricto, las XBAP suelen ofrecer una interface de navegación al usuario. Esto permite que la aplicación se comporte como un cliente Web que probablemente es lo que espera el usuario.

Debido a que las XBAP se cargan desde la Web y se ejecutan en un explorador, la seguridad de acceso a código de .NET Framework únicamente les otorga confianza limitada. Por esta razón, la capacidad de las XBAP no es tan amplia como la de aplicaciones de WPF independientes.

Entre las limitantes de aplicaciones tipo XBAP están las siguientes operaciones:

- Crear ventanas independientes.
- Mostrar cuadros de diálogo definidos por la aplicación.
- Mostrar el cuadro de diálogo Guardar iniciado desde XBAP.
- Tener acceso al sistema de archivos sin límite de espacio para almacenamiento aislado.
- Actuar como cliente de automatización de interface de usuario.
- Usar WCF(Windows Communication Foundation).

Las aplicaciones de WCF deben disponer de plena confianza, por lo que las XBAP implementadas desde Internet no son compatibles con esta tecnología; en su lugar, pueden hacer uso de los servicios Web ASP.NET, conocidos como

ASMX, para comunicarse con la aplicación Web a partir de la cual se realizó la carga.

Se puede crear páginas XAML puras, para Internet Explorer; que funcionen en conjunto con aplicaciones XBAP, pero es necesario considerar que XAML permite mostrar páginas estáticas en el explorador y para el control de eventos se requiere la creación de una XBAP.

Documentos XPS.- Son documentos de formato fijo en el entorno de WPF. Desempeñan una función clara en las interfaces de usuario. Para la presentación de este tipo de documentos WPF cuenta con el control “DocumentViewer”.

De forma conjunta con XAML los documentos XPS pueden incluir datos binarios como imágenes en formatos diversos, entre ellos: JPEG, PNG, TIFF y WMPhoto, datos de fuentes, información sobre la estructura de documentos, etc.

Este tipo de documentos no son específicos de Windows, en consecuencia la Corporación Microsoft tiene la intención de proporcionar visores de XPS, para otras plataformas como Apple Macintosh.

A partir de Windows Vista, XPS se convierte en formato de impresión nativo para Windows ya que actúa como lenguaje de descripción de páginas; las impresoras compatibles con XPS pueden representar directamente este tipo de documentos, aunque no sea lo primero que se espera de una interface de usuario, el uso generalizado de XPS ilustra el amplio margen que puede abarcar una tecnología como WPF.

Windows Presentation Foundation y Windows Forms.- Desde el lanzamiento inicial de .NET Framework, el uso de “Windows Forms” se ha convertido en una especie de estándar de desarrollo de interfaces de usuario debido a su amplia gama de controles que interactúan sin problema en aplicaciones multicapa.

El aparecimiento de WPF no pretende substituir en su totalidad el uso de este recurso que ofrece el entorno .NET; sino que plantea potenciar el uso de estos controles compartiendo en un mismo espacio de trabajo la funcionalidad de WPF con la versatilidad de Windows Forms; de esta forma es posible que los usuarios interactúen con cuadros de diálogo de WPF y de “Windows Forms” en la misma aplicación sin notar la diferencia.

Todo esto es posible gracias a elementos como “WindowsFormsHost”; control que permite alojar controles de “Windows Forms”, admitiendo además el alojamiento de controles de “ActiveX”; de modo que las aplicaciones de WPF

disponen de acceso a la gran biblioteca de controles existente basada en la anterior tecnología.

De forma similar, las aplicaciones de “Windows Forms” usan “ElementHost”, el que admite el alojamiento de controles, paneles y otros elementos de WPF.

Pese a la compatibilidad de componentes y controles entre las dos tecnologías; existen algunas restricciones como:

- No es posible colocar un control de WPF sobre un control de “Windows Forms”; este último siempre estará por encima.
- Los efectos de transparencia y transformaciones de WPF tampoco funcionan bien con los controles de “Windows Forms”.
- Los controles “WindowsFormsHost” y “ElementHost” requieren plena confianza, por lo que las aplicaciones de WPF que hagan uso de ellos no podrán ejecutarse como XBAP.

Aun así, un gran porcentaje de las aplicaciones de Windows pueden usar tanto WPF como “Windows Forms” para la creación de su interface de usuario.

Windows Presentation Foundation y Direct3D.- WPF depende por completo de Direct3D para todas las tareas de representación de sólidos y elementos visuales tridimensionales. Debido a esto WPF no substituye a Direct3D; sino que potencia esta herramienta en un entorno de desarrollo más concreto y de fácil uso.

“Direct3D es una herramienta esencial que los desarrolladores de Windows usan para crear gráficos tridimensionales y forma parte de la familia de API de DirectX de Microsoft.”⁴

Direct3D es una de las principales opciones para el desarrollo tridimensional, y es empleada en diversos tipos de aplicaciones ya sean estas de tipo técnico o juegos. WPF no cubre en su totalidad la funcionalidad de Direct3D; sino que emplea a Direct3D como soporte de desarrollo.

Las aplicaciones de WPF también pueden alojar código de Direct3D mediante la clase “HwndHost”. WPF y Direct3D desempeñan papeles únicos, por lo que las dos tecnologías forman parte esencial de la plataforma de Windows.

Windows Presentation Foundation y AJAX.- Los desarrolladores han encontrado favorable la compatibilidad generalizada con AJAX porque permite crear interfaces de usuarios más interactivas en la Web, sin embargo tiene la desventaja de no ser compatible con algunos tipos de contenidos por lo que

⁴ David Chappell & Chappell & Associates- Introducción a Windows Presentation Foundation

gráficos, vídeos, animaciones y otros estilos de interacción más modernos que no dependen únicamente de la compatibilidad con AJAX posiblemente se creen como XBAP.

2.2 Silverlight

Silverlight es una nueva tecnología de presentación Web, y subdivisión de lo que es “Windows Presentation Foundation”. Conocida también como WPF/E es decir “Windows Presentation Foundation Everywhere”. Se basa, de igual manera que WPF; en la tecnología XAML y trabaja además en conjunto con “JavaScript” y lenguajes como Visual Basic, C# y está orientada a interactuar incluso con ASP.NET y AJAX.

Silverlight ha sido creada para ser ejecutada en distintas plataformas ya que está orientada a entornos como exploradores Web e incluso dispositivos móviles de última generación. Esta tecnología es una herramienta que pretende alcanzar la funcionalidad y riqueza de contenido que ofrece WPF como tal. Y debido a que aún se encuentra en proceso de desarrollo, tiene deficiencias como la poca o nula inclusión e interacción con elementos como los componentes de “Windows Forms”.

Pese a esto, Silverlight posee en si elementos como: XPS, composición gráfica vectorial y aceleración de hardware; que permiten que el contenido Web de un

sitio o página pueda cargar elementos gráficos vectoriales, e incluso audio y video de una manera rápida, sencilla y bastante atractiva.

Además, al trabajar directamente con XAML su funcionalidad se potencia en gran manera ya que puede generar elementos reutilizables y de fácil modificación que pueden acoplarse sin problema a los lugares donde se los requiera.

2.2.1 Definición y Origen de Silverlight

El apareamiento de la tecnología Silverlight propone un entorno de trabajo flexible sustentado en XAML y plantea un nuevo estándar de aplicativos orientados a la Web a la vez que se fundamenta en la modularidad y polimorfismo que es parte de la programación orientada a objetos.

“A pesar de todas las cosas buenas que se pueden decir sobre HTML, CSS, e incluso JavaScript. En la actual generación de interfaces de usuario; no es suficiente contar únicamente con estas herramientas ya que su nivel de desarrollo visual es pobre en comparación con aplicativos desarrollados con herramientas como Adobe Flash.”⁵

Por citar un ejemplo; muchas técnicas y bibliotecas de JavaScript, se han desarrollado y compartido en diversas tecnologías y plataformas; pero ninguna de ellas ha sido absolutamente eficaz. Además de las marcadas diferencias entre browsers o exploradores, que no siempre son compatibles con las aplicaciones generadas en las diversas herramientas existentes.

⁵ Adam Nathan - Silverlight 1.0 Unleashed - SAMS

Dibujar líneas simples, o incorporar vídeo, entre otras cosas son tareas extremadamente complicadas o imposibles empleando solamente HTML. Esto no quiere decir que HTML es una tecnología deficiente o inútil; sino que su creación y su funcionalidad fue orientada con otro fin; es decir fue diseñada con un propósito basado principalmente en hipervincular documentos y archivos. Y debido al momento que fue creado HTML, no se considero mucho la presentación de los datos y mucho menos se considero aplicativos ricos en contenido visual; mucho menos contenido interactivo como el que proponen herramientas como Silverlight y Adobe Flash.

Es necesario mencionar que pese a que Adobe Flash, en la actualidad, se ha convertido prácticamente en un estándar para este tipo de aplicaciones; sin embargo conserva el mismo problema básico de HTML debido a que fue elaborado y orientado para animaciones y enlaces específicos de aplicativos multimedia y Web, sin considerar sistemas robustos que tengan interacción continua con Mainframes que soportan grandes tráfico de información; que requieren manejar estrictas políticas de seguridad e inclusive atender a una considerable cantidad solicitudes simultáneas; además de interactuar con bases de datos de gran escala.

Pese a que Silverlight se proyecta a ser una herramienta rica en contenido visual y funcional; aun posee algunas limitantes como lo es el trabajo con interfaces tridimensionales; elemento que ya está incluido en WPF nativo y

pueden ser implementado directamente desde la herramienta MSE Blend únicamente para aplicativos de “Windows Presentation Foundation” mas no en Silverlight.

2.2.2 Estructura de Silverlight

Silverlight, al ser derivada de “Windows Presentation Foundation”, emplea recursos de la infraestructura de programación de Windows conocida como .NET Framework a partir de su tercera generación. Y define su funcionalidad apoyándose en lenguajes de programación como “Java Script”, Visual Basic, C# y XAML.

Para lograr ser una tecnología destacada, Silverlight necesita ejecutarse correctamente en diversos sistemas operativos conocidos y en los exploradores más usados. Como resultado, la primera versión es compatible con los exploradores Firefox y Safari que se ejecutan en Mac OS X, así como en Firefox y en Internet Explorer ejecutándose en Windows.

La meta de Silverlight es posicionarse de ser posible como una tecnología de desarrollo universal, por lo tanto en futuras versiones de la tecnología se considera la compatibilidad y funcionalidad con más sistemas operativos y exploradores; incluyendo las principales plataformas de software libre.

Por otro lado, Silverlight también es completamente independiente y no depende de otros productos tales como Windows Media Player para la reproducción de vídeo ni de Microsoft .NET Framework 3.0 para el análisis de XAML.

Funcionamiento.- Al ser una aplicación que corre como un extra en la Web; es necesario descargar el módulo de compilación que ejecute el add-on Silverlight.

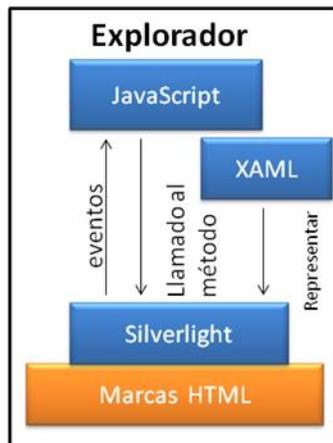
“Esencialmente, Silverlight es un complemento de explorador que representa XAML, exponiendo su modelo de objetos del documento (DOM) interno y el modelo de evento al explorador de manera traducible en secuencias de comandos.”⁶

Silverlight analiza el contenido de XAML en la inicialización, y después carga el contenido apropiado. De esta forma, los diseñadores pueden reunir un documento XAML que contenga gráficos, animaciones y escalas de tiempo; mientras que los desarrolladores pueden adjuntarlos al código de una página para implementar la funcionalidad.

Dado que XAML se basa en XML; el documento que define la Interface de Usuario que se descarga al cliente se basa en texto y, por lo tanto, no da problemas a motores de búsqueda ni firewalls. Además, XAML puede

ensamblarse y emitirse en tiempo de ejecución, mediante una aplicación de servidor; lo que no sólo ofrece una experiencia gráfica enriquecida, sino también personalizada y dinámica.

A continuación la figura 2.2, muestra la estructura de una aplicación Silverlight sencilla, usando un archivo XAML estático que define su Interface de Usuario y un archivo JavaScript para el control de eventos.



Estructura de Aplicaciones Silverlight

Figura 2.2

El explorador crea una instancia del complemento y, como la parte de este proceso, carga el archivo XAML. Los eventos de este archivo, tales como el clic de un botón, los captura el explorador y los procesa JavaScript.

2.2.3 Arquitectura de Silverlight

Fundamenta su interface de programación en la API de Java Script DOM. La figura 2.3 indica la composición de la arquitectura Silverlight segmentando por un lado el tiempo de Ejecución de presentación y los complementos de explorador.



Arquitectura de Silverlight

Figura 2.3

Esta arquitectura administra los procesos a ejecutarse mediante eventos controlados por XAML, respondiendo según sea el caso a procesos de carga o incluso secuencias de animación completas.

Mediante métodos es posible invocar a animaciones u operaciones, iniciándolas o deteniéndolas desde el archivo XAML al archivo JavaScript, el mismo que puede contener una o varias operaciones relacionadas directamente con los elementos de la composición gráfica de Silverlight.

“El analizador; crea el DOM de XAML en memoria; el mismo que usará el núcleo de presentación y que controla la representación de los gráficos y las animaciones definidos por XAML. Además, el tiempo de ejecución contiene los códecs necesarios para la reproducción de contenido multimedia WMV, WMA y MP3.”⁷

La arquitectura de Silverlight contiene además la presentación del núcleo de un framework, el cual es responsable de la interoperabilidad y el ingreso de datos por parte del usuario, los gráficos, animación, la reproducción de medios, el soporte a la administración de derechos digitales y la integración del DOM. Sus componentes a más de XAML son los siguientes:

Input : Maneja la información que aportan los dispositivos de entrada como los teclados, el ratón, el stylus, etc.

⁷ Alfredo Ceballos – Artículo: “Qué es Silverlight?” www.silverlight.org.mx

Núcleo UI: Maneja el rendimiento de las imágenes bitmap (incluyendo las imágenes raster como JPEG, los gráficos vectoriales, el texto y las animaciones).

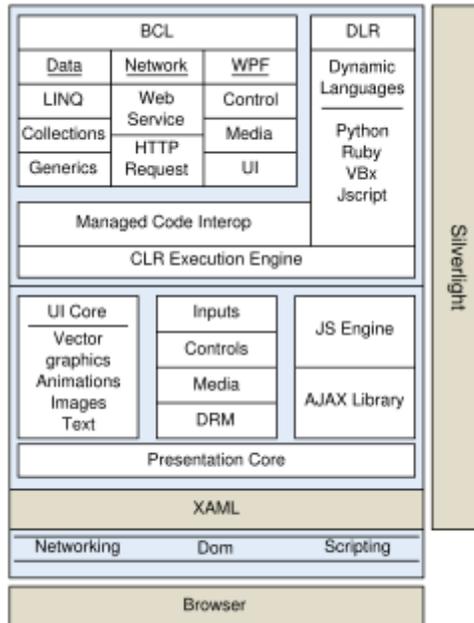
Media: Reproducción en línea de MP3, Windows Media y VC-1.

Una aplicación de Silverlight 1.0 comienza por invocar el control de Silverlight mediante una la página HTML, donde se carga el archivo XAML, que contiene un objeto “Canvas”, el cual actúa como un armario para otros elementos.

Silverlight provee varios símbolos geométricos básicos como líneas, elipses y otros, controles como texto, imágenes y multimedia, los mismos están propiamente posicionados para alcanzar la disposición deseada. De modo que estos elementos pueden ser animados usando el reproductor de eventos; algunos efectos de animaciones están predeterminados, mientras que otros pueden ser compuestos de otros efectos pre-definidos. Eventos como el movimiento del teclado o el ratón pueden ser manejados por scripts personalizados.

“La manipulación programática del UI se obtiene al usar lenguajes de scripts para modificar el DOC del objeto Canvas del Silverlight. Para facilitar esto, Silverlight muestra una API DOM, accesible desde cualquier lenguaje de scripts soportado por Silverlight.”⁸

La figura 2.4 muestra la estructura funcional de Silverlight:



Estructura Funcional de Silverlight

Figura 2.4

2.2.4 Propiedades y Componentes de Silverlight

A más de la propiedad que regula el ancho y el alto de un elemento definido mediante XAML en Silverlight; existe otras propiedades que son parte de la tecnología y estas son:

Canvas.- Una etiqueta de este tipo puede tener uno o más elementos secundarios, incluido dentro de estos elementos, se pueden crear otros. Los elementos secundarios de una etiqueta Canvas tienen posiciones relativas respecto a su elemento principal, no al elemento raíz.

Background.- Es el repositorio de todo el contenido cargado por Silverlight, siendo más poderoso que el modelo que emplea HTML para montar imágenes y colores. La propiedad “Background” permite controlar a más de la cromática de fondo de una pantalla efectos como degradación de color, transparencias manejando el mismo esquema de color RGB (red, green, blue).

IsWindowsless.- Es una propiedad única de Silverlight que permite un manejo de elementos dinámicos que pueden pasar a un segundo plano. Esta propiedad tiene dos tipos de comportamientos que son:

- En HTML, es posible montar contenido de Silverlight sobre HTML y viceversa.
- Permite el manejo de transparencias a modo de máscaras.

InplaceInstallPrompt.- Es la herramienta que invoca a los motores de Silverlight para cargar contenido. En el caso de no existir dichos motores, se encarga de retribuir la información necesaria para solicitar la descarga de dicho contenido, de modo sea cargado en la mejor versión existente para la aplicación Silverlight.

MaxFramerate.- Permite manipular el número de frames o cuadros por segundo. Por defecto Silverlight define 24 frames por segundo. Este valor puede cambiar de acuerdo a la necesidad del desarrollador.

Eventos.- Silverlight por defecto emplea un control de errores mediante el uso de código JavaScript que ya viene definido de esa forma. Estos dos eventos controlados son onError y onLoad.

2.2.5 Herramientas para Silverlight

Herramientas de Desarrollo

Para el área de desarrollo se recomienda emplear

- Microsoft Visual Studio 2005 o Microsoft Visual Studio 2008 (nombre código VS Orcas)
- Microsoft Web Developer 2008 Express Edition.

Estas Suites actualmente contienen componentes nativos de la herramienta además de utilitarios auxiliares que permiten efectuar un desarrollo de aplicaciones para WPF y Silverlight de manera eficiente. Cabe mencionar que para algunas versiones de Visual Studio será necesario descargar el editor de XAML (“XAML editor”); y se debe considerar que la versión de Java Editor incluida en versiones como VS 2005; no es muy rica en contenido. Además se requiere descargar el Framework 3.0 o superiores.

Por su parte Visual Studio 2008 incluye ya varias mejoras para el desarrollo de aplicaciones Silverlight, incluyendo el Framework 3.0 ó a su vez el .NET Framework 3.5; de acuerdo a la versión de Visual Studio; además de

herramientas como “Intellisense” para auto completado de código y “debugging”.

También se puede utilizar para el proceso de desarrollo editores como:

- Block de notas o editores de texto
- Dream Weaver (Adobe)

Herramientas de Diseño

Para diseño de interfaces con Silverlight; se recomienda utilizar la suite de Microsoft Expression que se define a continuación.

Microsoft Expression Studio.- Es la suite de la corporación Microsoft, orientada al desarrollo de aplicaciones multimedia y de diseño lanzada oficialmente por Microsoft a finales de abril del 2007. Esta suite de productos, ha sido diseñada para ser funcional con la tecnología WPF. Esta compuesta por:

Microsoft Expression Web.- Bajo el nombre código Quartz. Es un software para desarrollo Web (“WYSIWYG”) y editor HTML. Es la herramienta sucesora de Microsoft FrontPage, con la cualidad que es mucho más completa que su antecesora, incluyendo entre sus principales funcionalidades su propia herramienta para rendereo de imágenes, en lugar del motor de capas Trident

de Internet Explorer. Como tal contiene mejoras que corrigen las deficiencias de Front Page.

Microsoft Expression Media.– Herramienta desarrollada para administrar medios y complementos digitales. Ha sido desarrollada basándose en iView Media.

Microsoft Expression Blend (Sparkle).– Constructor visual para Interfaces de Usuario orientadas a aplicaciones de WPF y derivados. Permite la generación de interfaces de usuario ricas en contenido tanto para aplicaciones de escritorio como para aplicaciones Web. Ha sido desarrollada usando el .NET Framework 3.0 y WPF. Las aplicaciones generadas son basadas en XAML.

Microsoft Expression Design (Acrylic).– Editor gráfico para construcción de elementos vectoriales y rasterizados. Fue creada en su totalidad empleando WPF y su estructura fue hecha en base de Creature House Expression.

Microsoft Expression Encoder (VC-1).– Generador para proceso de ENCODE orientado a XAML. Actualmente solo puede ser usado en el sistema operativo Windows y es la herramienta más nueva de esta suite. Se emplea para codificación de contenido de audio y video digital; permitiendo comprimir archivos a VC-1. Silverlight soporta reproducción de contenido VC-1 de video

en todos los navegadores compatibles sin requerir el control ActiveX del Reproductor de Windows Media.

Otras Herramientas.- Pese a ser WPF y Silverlight una tecnología nueva; existen ya herramientas que proponen soluciones para implementar WPF y XAML. Este es el caso de:

Microsoft Cider.- Add In para visual Studio 2005 diseñado para desarrollar WPF.

Electric Rain ZAM 3D.- Herramienta para construcción de sólidos y modelos en 3D, compatibles con XAML. Aún se encuentra en versiones de prueba.

Mobiform Aurora.- Permite la construcción de interfaces únicamente para WPF no para Silverlight.

Xceed DataGrid for WPF.- Generador de controles tipo datagrids únicamente para WPF no para Silverlight.

2.2.6 Requerimientos de Software para ejecución de contenido Silverlight

Para reproducir contenido Silverlight se requiere:

1. Versión 1.0 ó 1.1 de MS Silverlight.

2. Sistema Operativo Windows XP, Windows Vista, Mac OSX Tiger ó Leopard (No está disponible para IA64 Server versión, ó Windows 2000).
3. Web Browser como Internet Explorer 6.0, Firefox 1.5, Safari 2.0.4; ó sus versiones superiores.

Para el caso de exploradores para Linux, se debe tener en cuenta que Silverlight aún se encuentra en período de prueba por lo tanto no se aconseja su uso; pese a eso Silverlight es compatible con:

- Moonlight,
- Novell (implementación “open source” de Silverlight)
- Firefox
- Opera
- Konqueror

2.2.7 Lenguajes que interactúan con Silverlight

2.2.7.1. JavaScript

JavaScript es el lenguaje de programación empleado en una gran parte de los sitios Web existentes. Su propósito es añadir interactividad a los “browsers” y a los portales o sitios en Internet. Por otro lado es un lenguaje empleado como complemento de ASP.NET, a tal punto que muchos controles de esta tecnología han sido implementados usando este lenguaje, además es de fácil aprendizaje y versátil en su aplicación.

Un factor importante que es necesario considerar es que JavaScript es la base que se ha empleado para el desarrollo e implementación de AJAX.

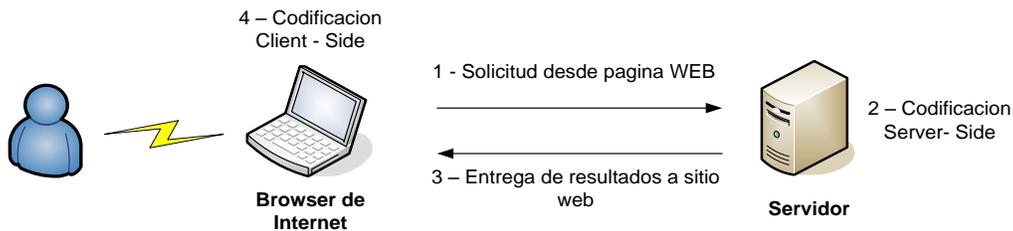
Funcionamiento de JavaScript.- Como se menciono anteriormente; al haber sido diseñado con un propósito diferente, Java Script tiene marcadas limitantes que lo vuelve un lenguaje estático.

Por citar un ejemplo de esta limitación. Solo hace falta considerar el caso de implementar un reloj de sistema que muestre la hora y la fecha actual en un sitio HTML. Esta tarea es prácticamente imposible ya que sería necesario actualizar este reloj de manera manual, y es ilógico imaginar un reloj que no muestre la hora en tiempo real.

JavaScript soluciona esto mediante Aplicaciones Cliente – Servidor. Este tipo de aplicaciones se asemejan bastante a las transacciones que se dan en la vida cotidiana en el mundo tangible; es decir, una persona tiene el estado de “cliente” cuando se acerca a un lugar determinado en búsqueda de un bien o servicio. De igual manera en Internet cuando se ingresa a un portal o sitio determinado en búsqueda de algo y se efectúa una consulta o un requerimiento determinado; el estado de quien ingresa pasa a ser el de “cliente” siendo el servidor quien responde a las solicitudes que el cliente efectuó.

Java Script es un lenguaje orientado a este tipo de aplicaciones Cliente – Servidor; por tanto será necesario considerar su funcionamiento y la manera en la que este lenguaje actúa ante la interacción de Cliente y Servidor. Para esto hay que dividir al proceso en dos partes que son “ Client - Side” y “Server - Side”. Es decir el lado del cliente y el lado del servidor por separados.

Para entender esto de una mejor manera, la figura 2.5 ilustra esta interacción.



Aplicaciones Cliente Servidor

Figura 2.5

Una vez que se ha ingresado a la dirección electrónica requerida, o se ha ingresado mediante un “link” determinado; se ejecuta una solicitud a través del vínculo de la página. Luego de esto el servidor busca el contenido de esta página; abre y ejecuta el código implementado en la misma. Este código puede incluso contener código ASP.NET.

Cuando se ha ejecutado este código, el resultado es redireccionado al navegador. El resultado es una página que contiene HTML y código CSS y opcionalmente, código del lado del cliente. Si existe este tipo de código siendo

este JavaScript, el browser lo procesará y lo desplegará al usuario y va correr cuantas veces sea necesario mientras la página este siendo desplegada en el navegador.

Sintaxis.- La sintaxis de JavaScript es muy similar a la de otros lenguajes como Java y C. Las normas básicas que definen la sintaxis de JavaScript son las siguientes:

- No se tienen en cuenta los espacios en blanco y las nuevas líneas. Como sucede con XHTML, el intérprete de JavaScript ignora cualquier espacio en blanco, por lo que el código se puede ordenar de forma adecuada para su manejo (tabulando las líneas, añadiendo espacios, creando nuevas líneas, etc.)
- Se distinguen las mayúsculas y minúsculas: al igual que sucede con la sintaxis de las etiquetas y elementos XHTML. Sin embargo, si en una página XHTML se utilizan indistintamente mayúsculas y minúsculas, la página se visualiza correctamente, siendo el único problema la no validación de la página. Sin embargo, si en JavaScript se intercambian mayúsculas y minúsculas el script no funciona.

- No se define el tipo de las variables, no es necesario indicar el tipo de dato que almacenará. De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.
- No es necesario terminar cada sentencia con el símbolo “;”. En la mayoría de los lenguajes de programación, es necesario terminar cada sentencia con este símbolo. JavaScript permite no hacerlo, pero es muy recomendable seguir la tradición de terminar las sentencias con “;”.
- Se pueden incluir comentarios, se utilizan para añadir alguna información referente al código fuente del programa.

2.2.7.2. ASP.NET

Actualmente, ASP.NET es una de las tecnologías empleadas con mayor frecuencia por una gran parte de desarrolladores Web. Como tal es una tecnología mucho más completa y robusta que la versión antecesora llamada ASP o “Active Server Pages”.

ASP.NET, es una plataforma de desarrollo Web unificada que proporciona los servicios necesarios a los desarrolladores para generar aplicaciones Web a nivel empresarial y trabaja en conjunto con el Microsoft .NET Framework.

Posee un marco de trabajo de programación generado en “Common Language Runtime”; que puede utilizarse en un servidor para generar aplicaciones Web, ofreciendo varias ventajas importantes acerca de los modelos de programación Web anteriores.

Su arquitectura por su parte permite crear aplicaciones interactivas de primer nivel incluyendo además muchas características nuevas; como lo es un modelo unificado de desarrollo WEB, que incluye los servicios necesarios para desarrolladores respecto a herramientas para la construcción de aplicaciones Web de tipo “Build Enterprise – class”.

Características de ASP.NET

Conforme a lo detallado en el “Curso de ASP.NET 3.5”⁹: se describen las características de este lenguaje de la siguiente forma:

Mejor Rendimiento.- A diferencia de sus predecesores, ASP.NET puede aprovechar las ventajas del enlace anticipado, la compilación “just-in-time”, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento notable en rendimiento; incluso antes de ensamblar código.

⁹ Curso de ASP.NET 3.5 - www.adrformacion.com/cursos/aspnet3/leccion2/tutorial7.html

Compatibilidad con herramientas de primer nivel.- El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (Integrated Development Environment, IDE) de Visual Studio. La edición “WYSIWYG”, los controles de servidor de arrastrar y colocar y la implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta.

Compatibilidad con varios lenguajes.- La Plataforma Microsoft .NET ofrece actualmente compatibilidad integrada para tres lenguajes: C#, Visual Basic y JScript. En ASP.NET 2.0 se pueden utilizar lenguajes diferentes en la misma aplicación debido a que ASP.NET genera varios ensamblados independientes, para cada lenguaje.

Eficacia y flexibilidad.- Gracias a la interoperabilidad de “Common Language Runtime”; la eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de .NET Framework, la mensajería y las soluciones de acceso a datos se encuentran accesibles en la Web de manera uniforme. ASP.NET es también independiente del lenguaje, por lo que puede elegir la codificación que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes.

Simplicidad.- ASP.NET facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la

implementación y la configuración de sitios, ya que el marco de trabajo de una página de ASP.NET permite generar interfaces de usuario, que separan claramente la lógica de aplicación del código de presentación y control de eventos en un sencillo modelo de procesamiento de formularios de tipo Visual Basic.

Además, Common Language Runtime simplifica la programación, con servicios de código administrado como el recuento de referencia automático y el recolector de elementos no utilizados.

Facilidad de uso.- ASP.NET emplea un sistema de configuración jerárquico basado en texto que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web. Debido a que la información de configuración se almacena como texto sin formato, se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local. Esta filosofía de "administración local cero" se extiende asimismo a la implementación de las aplicaciones ASP.NET Framework. Estas se implementan en un servidor mediante la copia de los archivos necesarios al mismo. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.

Implementación flexible.- Dado que ASP.NET compila su sitio Web la primera vez que un usuario lo solicita, simplemente puede copiar el código

fuente de su aplicación en el servidor Web de producción. Sin embargo, ASP.NET también proporciona opciones de pre-compilación que permiten a un sitio Web interpretarlo antes de haberlo implementado o compilarlo una vez implementado pero antes de que un usuario lo solicite.

Esta pre-compilación tiene varias ventajas. Puede mejorar el rendimiento de un sitio Web cuando recibe la primera solicitud porque no se producirá ningún retraso mientras ASP.NET compila el sitio. Pre-compilar también puede ayudarle a encontrar errores que solo se encontrarían cuando un usuario solicita una página. Finalmente, si se pre-compila el sitio Web antes de implementarlo, es posible implementar los ensamblados en lugar del código fuente.

ASPNET_Compiler.exe es la herramienta que proporciona las opciones de pre-compilación siguientes:

- Compilación en contexto
- Pre compilación completa no actualizable
- Pre compilación actualizable

Escalabilidad y Disponibilidad.- Se ha diseñado ASP.NET teniendo en cuenta la escalabilidad con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores.

Además, el motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno de estos procesos no se comporta adecuadamente por filtraciones o bloqueos; se puede crear uno nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.

Posibilidad de personalización y extensibilidad.- ASP.NET presenta una arquitectura que permite insertar el código en el nivel adecuado. De hecho, es posible extender o reemplazar cualquier subcomponente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado.

Seguridad.- Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad en las aplicaciones. ASP.NET ofrece más control para implementar políticas de seguridad en las mismas que funcionan en conjunción con la seguridad de Microsoft Internet Information Services (IIS) e incluye servicios de autenticación y autorización para implementar el modelo de seguridad de ASP.NET.

También incluye una característica de seguridad basada en funciones que se puede implementar para cuentas de usuario de Microsoft Windows y que no son de Windows.

Sintaxis.- Aunque algunos de los elementos de la sintaxis de ASP.NET son bastante similares al lenguaje ASP inicial: otros son exclusivos del nuevo marco de trabajo desarrollado específicamente para la tecnología .NET.

Herramientas y Funcionalidad de ASP.NET

Formularios Web.- También llamados “Web Forms”; son el marco de trabajo de la página de formularios Web de ASP.NET. Posee un modelo de programación escalable de “Common Language Runtime” que puede utilizarse en el servidor para generar páginas Web dinámicamente.

En particular, proporciona, la capacidad para crear y utilizar controles de la interfaz de usuario reutilizables que puedan encapsular funcionalidades comunes y, así, reducir la cantidad de código al escribir una página.

Por otro lado provee la funcionalidad para estructurar limpiamente la lógica de la página de forma ordenada. Además de la capacidad para que las herramientas de desarrollo proporcionen un fuerte soporte de diseño “WYSIWYG” es decir; lo que ve es lo que se imprime a las páginas.

Los “Web Forms” de ASP.NET consisten en archivos de texto con una extensión de nombre de archivo .aspx, que pueden implementarse por todo un árbol de directorio raíz virtual para “IIS”.

Cuando un explorador cliente solicita recursos .aspx, el motor en tiempo de ejecución de ASP.NET analiza y compila el archivo de destino en una clase de .NET Framework.

Controles de servidor.- Son desarrollados para proporcionar un modelo de programación estructurado para el acceso a las propiedades, los métodos y los eventos de los controles de la interfaz de usuario desde el código de servidor. Además de los controles integrados, el marco de trabajo de páginas ASP .NET; permite crear controles de usuario y incluso personalizar controles.

Administración de estados.- HTTP como tal; es un protocolo sin estado. Las solicitudes se atienden conforme llegan; similares a un arreglo tipo “cola”; cuando una solicitud ha sido procesada, todos los datos se descartan. No se mantiene el estado entre las solicitudes, ni siquiera cuando proceden del mismo cliente. Sin embargo, para determinadas soluciones, es muy útil mantener el estado entre solicitudes. ASP.NET permite mantener el estado de la aplicación y el estado de la sesión mediante el uso de variables de la aplicación y de la sesión respectivamente.

Almacenamiento en caché.- La memoria caché de ASP.NET es general para aplicaciones Web, proporciona tanto una interfaz simple para el almacenamiento en caché y una interfaz más avanzada que expone servicios de dependencia de caducidad y cambio.

Enlace de datos.- Permite enlazar componentes con orígenes de datos, así como simples propiedades, colecciones, expresiones y métodos; brindando mayor flexibilidad cuando se utilizan datos desde una base de datos o desde otros medios.

Módulos HTTP y controladores HTTP.- Forman parte integral de la arquitectura de ASP.NET. Cada solicitud de ASP.NET es procesada por varios módulos HTTP y después es procesada por un único controlador HTTP. Después de que este haya procesado la solicitud, ésta es devuelta a través de los módulos HTTP. Un ejemplo de esto, es el módulo de autenticación y el módulo de sesión.

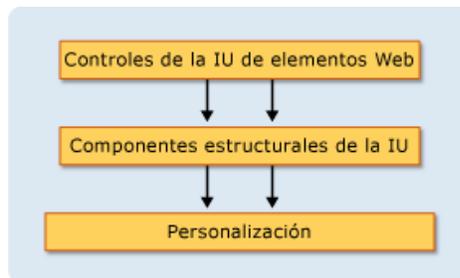
Conjunto de controles de elementos Web.- Es un grupo de componentes que funcionan conjuntamente para permitir la creación de páginas Web en las que los usuarios finales pueden modificar el aspecto y el comportamiento de la interfaz de usuario (IU) directamente desde un explorador.

Categorías de los componentes de los elementos Web.- El conjunto de controles de elementos Web está formado por tres unidades de creación fundamentales:

- las posibilidades de personalización,

- los componentes estructurales de la IU necesarios para la utilización de los controles que son elementos Web en una página,
- los controles de la IU de los elementos Web propiamente dichos.

La figura 2.6 muestra las relaciones entre estas unidades de creación del conjunto de controles de elementos Web y la jerarquía de controles de elementos Web.



Relación de controles Web con jerarquía de elementos web

Figura 2.6

La personalización es la base de las características de los elementos Web debido a que permite a los usuarios personalizar el diseño, aspecto y comportamiento de los controles de los elementos Web en una página.

La configuración personalizada se mantiene durante la sesión de explorador actual e incluso en el almacenamiento a largo plazo, de forma que la configuración de un usuario se guarda para sesiones de explorador futuras.

Los componentes estructurales de la interfaz de usuario dependen de la personalización y proporcionan la estructura y los servicios centrales que todos los controles de los elementos Web necesitan.

Un componente estructural de la Interface de Usuario y necesario en todas las páginas de elementos Web; es el control WebPartManager. Aunque nunca está visible, este control tiene la tarea esencial de coordinar todos los controles de los elementos Web de una página.

El segundo tipo de componente estructural de la Interface de Usuario; es “la zona”. Las zonas actúan como administradores de diseño en una página de elementos Web. Contienen y organizan los controles derivados de la clase Part (controles de elementos), y proporcionan la capacidad de realizar un diseño de página modular, con orientación horizontal o vertical.

Las zonas proporcionan además elementos de la IU comunes y coherentes como el estilo del encabezado y el pie de página, el título, el estilo de borde, los botones de acción, etc., para cada uno de los controles que contienen.

Estos elementos comunes se denominan el cromo de un control. Hay varios tipos de zonas especializados que se utilizan en distintos modos de presentación y con distintos controles.

Además de crear controles de elementos Web personalizados, se puede utilizar los controles de servidor ASP.NET existentes, los controles de usuario o controles de servidor personalizados como controles de elementos Web.

Ciclo de vida de página.- Cuando se ejecuta una página ASP.NET, ésta recorre un ciclo de vida en el que realiza una serie de pasos de procesamiento. Entre ellos se incluyen la inicialización, la creación de instancias de controles, la restauración y el mantenimiento del estado, la ejecución del código del controlador de eventos y la representación.

Es necesario entender el funcionamiento del “ciclo de vida de página”; para poder implementar de manera efectiva, tanto el código, como el uso de controles personalizados; inicializarlos, asignar datos de estado de vista a las propiedades de los controles y ejecutar la lógica de comportamiento de éstos correctamente.

El ciclo de vida de un control está basado en el de la página, pero ésta produce en un control más eventos de los que están disponibles para la página ASP.NET por sí sola.

Fases generales del ciclo de vida de la página.- Además de las fases del ciclo de vida de la página, existen las fases de la aplicación que se producen antes y después de una solicitud pero no son específicas de una página.

En términos generales, la página recorre las fases descritas en la tabla siguiente.

Fase	Descripción
Solicitud de página	Se produce antes de que comience el ciclo de vida de la página. Cuando un usuario solicita la página, ASP.NET determina si ésta se debe analizar y compilar (a fin de que comience el ciclo de vida de la página) o si se puede enviar una versión en caché de la página como respuesta sin ejecutar la página.
Inicio	Se establecen las propiedades de la página, como Request y Response. En esta fase, la página también determina si la solicitud es una devolución de datos o una nueva solicitud, y establece la propiedad IsPostBack. Además, durante esta fase se establece la propiedad UICulture de la página.
Inicialización de página	Durante la inicialización de la página, los controles incluidos en ella están disponibles y se establece la propiedad UniqueID de cada uno de ellos. Además, se aplican los temas correspondientes a esta. Si la solicitud actual es una devolución de datos, los datos de devolución aún no se han cargado y los valores de las propiedades del control no se han restaurado a los valores del estado de vista.
Carga	Si durante la carga; la solicitud actual es una devolución de datos, las propiedades del control se cargan con información recuperada del estado de vista y del estado del control.
Validación	Durante la validación, se llama al método Validate de todos los controles de validación, que establece la propiedad IsValid de cada uno de los controles de validación y de la página.

Control de eventos de devolución de datos	Si la solicitud es una devolución de datos, se llama a los controladores de eventos.
Representación	Durante la representación, el estado de vista se guarda en la página y, a continuación, ésta llama a cada uno de los controles para que aporte su salida representada al valor OutputStream de la propiedad Response de la página.
Descarga	Se llama a la descarga cuando la página se ha representado completamente, se ha enviado al cliente y está lista para ser descartada. Llegado este momento, se descargan las propiedades de la página, como Response y Request, y se llevan a cabo las operaciones de limpieza correspondientes.

2.3 XAML

El acrónimo XAML originalmente significaba “Extensible Avalon Markup Language” o en su definición en castellano “Lenguaje Extensible de Formato de Avalon”; habiendo sido Avalon el nombre clave original de WPF.

Este Lenguaje Extensible de Formato para Aplicaciones; es de tipo declarativo basado en XML que define los objetos y sus propiedades, usando esquemas para la programación declarativa de aplicaciones.

XAML es un lenguaje declarativo basado en XML y es la base de Windows Presentation Foundation y sus derivados como Silverlight. Fue diseñado para

soportar las clases y métodos de la plataforma de desarrollo .NET, optimizando la descripción grafica de interfaces de usuarios.

2.3.1 Funcionamiento

XAML permite crear elementos de la interface de usuario visibles; y a continuación separa su definición de la lógica en tiempo de ejecución, utilizando archivos de código subyacente que se unen al marcado mediante definiciones de clases parciales. De esta forma simplifica la creación de interfaces para el modelo de programación de .NET Framework.

A diferencia de la mayoría de los demás lenguajes de marcado, representa directamente la creación de instancias de objetos administrados. Este principio de diseño general habilita el código simplificado y el acceso a la depuración para los objetos que se crean en el lenguaje.

La capacidad de combinar código con marcado en XAML es importante porque XML por sí solo es declarativo, y no sugiere realmente un modelo para el control de flujo, debido a que es muy intuitivo para crear interfaces que van desde el prototipo a la producción.

El ejemplo 2.1 descrito a continuación; muestra la pequeña cantidad de código, necesaria para crear un botón como parte de una interfaz de usuario.

```
<StackPanel>  
<Button Content="Click Me"/>  
</StackPanel>
```

Código para la creación de un botón en XAML

Ejemplo 2.1

El botón creado tiene una presentación visual predeterminada procedente de los estilos del tema y un comportamiento predeterminado de su diseño de clase.

La información detallada a continuación tiene el respaldo teórico de la información técnica publicada en el sitio oficial de MSDN¹⁰, en lo referente a “Información general sobre XAML”.

Elementos de objeto XAML.- Este lenguaje de marcado, tiene un conjunto de reglas que asignan elementos de objeto a clases o estructuras, atributos a propiedades o eventos, y espacios de nombres de XML a espacios de nombres de CLR.

Los elementos de XAML se asignan a los tipos de Microsoft .NET, tal y como se definen en los ensamblados a los que se hace referencia y los atributos se asignan a los miembros de dichos tipos.

¹⁰ "Información General sobre XAML" – www.msdn.microsoft.com

En el ejemplo anterior se han especificado dos elementos de objeto: `<StackPanel>` (con una etiqueta de cierre) y `<Button/>`. Cada una de las cadenas `StackPanel` y `Button` se asignan al nombre de una clase definida por WPF y que forma parte de los ensamblados de WPF.

Al especificar una etiqueta de elemento de objeto se crea una instrucción para el procesamiento de XAML, que indica que se debe crear una nueva instancia de la clase con nombre al cargar la página XAML.

Cada instancia se crea llamando al constructor predeterminado de la clase subyacente, almacenando el resultado. Para que se pueda utilizar como un elemento de objeto en XAML, dicha clase o estructura debe exponer un constructor público predeterminado sin parámetros.

Definición de propiedades.- Las propiedades en XAML se establecen definiéndolas en un elemento de objeto empleando una sintaxis determinada. Las sintaxis que se pueden utilizar para una propiedad variarán, dependiendo de las características que se establece.

Para agregar características a los elementos de objeto, se establece los valores de las propiedades; el estado inicial de la instancia de objeto subyacente para un elemento de objeto está basado en el comportamiento del

constructor predeterminado. Normalmente la aplicación utilizará algo distinto de una instancia completamente predeterminada de cualquier objeto dado.

2.3.2 Sintaxis

Sintaxis de atributo.- En XAML, las propiedades se pueden expresar a menudo como atributos; por tanto, la sintaxis de atributo se diferencia del establecimiento de propiedades al ser más optimizada. En el ejemplo 2.2 se crea un botón que tiene texto rojo y un mensaje

```
<Button Background="Blue" Foreground="Red" Content="Este es un botón"/>
```

Código para la creación de un botón con atributos en XAML

Ejemplo 2.2

Sintaxis de elementos de propiedad.- En algunas propiedades de un elemento de objeto, no es posible usar la sintaxis de atributo, ya que el objeto o la información necesaria para proporcionar el valor de la propiedad no se puede expresar correctamente como una cadena simple.

En estos casos, se puede utilizar otra sintaxis conocida como sintaxis de elementos de propiedad, la cual establece la propiedad del elemento contenedor a la que se hace referencia en el contenido de la etiqueta.

Generalmente, el contenido es un objeto del tipo que la propiedad toma como valor propio, con la instancia del valor especificada como otro elemento de objeto.

La sintaxis para el elemento de propiedad en sí se define escribiendo el nombre de la propiedad como lo muestra el ejemplo 2.3

```
<nombreDeTipo.Propiedad>
```

Definición de elemento de propiedad

Ejemplo 2.3

Después de especificar el contenido, se debe cerrar el elemento de propiedad con una etiqueta de cierre como cualquier otro elemento. En las propiedades en las que se admite tanto la sintaxis de atributo como la de elementos de propiedad, ambas tienen normalmente el mismo resultado, aunque algunos detalles como el control del espacio en blanco pueden variar ligeramente entre ellas.

El ejemplo 2.4 demuestra cómo se establecen propiedades utilizando la sintaxis de “elementos de propiedad” para el elemento Button.

```

<Button>
  <Button.Background>
    <SolidColorBrush Color="Blue"/>
  </Button.Background>
  <Button.Foreground>
    <SolidColorBrush Color="Red"/>
  </Button.Foreground>
  <Button.Content>
    Este es un botón
  </Button.Content>
</Button>

```

Definición de elemento de propiedad

Ejemplo 2.4

En XML al definir <nombreDeTipo.Propiedad>, solamente se representa otro elemento, sin ninguna relación necesariamente implícita con un elemento primario TypeName, aparte del hecho de ser un elemento secundario.

En XAML, <nombreDeTipo.Property> implica directamente que “Property” es una propiedad de nombreDeTipo, establecida por el contenido del elemento de propiedad, y nunca será un elemento con nombre similar sino un elemento independiente con un punto en el nombre.

Propiedades y herencia de clases.- Aparecen como atributos de XAML en un elemento de WPF; se heredan a menudo de las clases base, por ejemplo, la propiedad Background no es una propiedad declarada inmediatamente en la clase Button, en su lugar, Background se hereda de la clase Control base. El comportamiento de la herencia de clases de los elementos XAML es otra diferencia significativa de la interpretación de XML.

La herencia de clases, especialmente cuando las clases base intermedias son abstractas, es una de las razones por las que el conjunto de elementos de XAML, y sus atributos permitidos son casi imposibles de representar correcta y completamente utilizando únicamente los tipos de esquema que se usan de rutina para la programación en XML, como los formatos DTD o XSD.

Valores de referencia y extensiones de marcado.- Las extensiones de marcado son un concepto de XAML. En la sintaxis de atributo, el símbolo de llaves { y } indican un uso de la extensión de marcado. Este uso hace que el procesamiento de XAML se aparte del tratamiento general de valores de atributo como una cadena literal o un valor directamente convertible en cadena.

Cuando las propiedades toman un valor de tipo de referencia, requerirán a menudo una sintaxis de elementos de propiedad, lo que crea siempre una nueva instancia, o una referencia de objeto mediante una extensión de marcado.

El uso de una extensión de marcado puede devolver una instancia existente, por lo que puede ser más versátil o incurrir en una menor sobrecarga de objetos.

Cuando se utiliza una extensión de marcado para proporcionar un valor de atributo, éste debería ser proporcionado por la lógica de la clase de respaldo para la extensión de marcado pertinente.

Además, permiten utilizar la sintaxis de atributo para proporcionar valores de referencia a las propiedades, aun cuando estas no admitan una sintaxis de atributo directa, o permiten habilitar un comportamiento concreto que difiera del comportamiento general que requiere que las propiedades de XAML se rellenen con valores del tipo de la propiedad.

Tipos de colección y propiedades de la colección XAML.- Este lenguaje especifica una característica por la que el elemento de objeto que representa un tipo de colección se puede omitir deliberadamente en el marcado. Cuando un procesador de XAML controla una propiedad que toma un tipo de colección, se crea implícitamente una instancia del tipo de colección adecuado, aun cuando el elemento de objeto para dicha colección no esté presente.

2.3.3 Extensiones de marcado específicas de WPF

Las extensiones de marcado que admiten referencias de recursos comúnmente utilizadas en la programación de WPF son:

StaticResource.- Genera un valor para una propiedad XAML sustituyendo el valor de un recurso ya definido.

DynamicResource.- Permite un valor para una propiedad XAML aplazando ese valor para que sea una referencia a un recurso en tiempo de ejecución. Una referencia de recurso dinámica fuerza una nueva búsqueda cada vez que se tiene acceso a este tipo de recurso.

Binding.- Proporciona un valor enlazado a datos para una propiedad según el contexto de datos que se aplique al elemento. Esta extensión de marcado es relativamente compleja, porque habilita una sintaxis fundamentalmente incluida en el propio código para especificar un enlace de datos.

RelativeSource.- Da información de origen para un objeto Binding que puede navegar por varias posibles relaciones en el árbol de elementos en tiempo de ejecución. Esto proporciona una fuente especializada para los enlaces que se crean en plantillas de uso múltiple o se crean en código sin conocimiento completo del árbol de elemento circundante.

TemplateBinding.- Permite que una plantilla de control utilice valores para propiedades con plantilla procedentes de propiedades definidas por el modelo de objetos de la clase que utilizará la plantilla.

2.3.4 Procesadores de XAML

Se define como cualquier programa que pueda aceptar XAML como lenguaje según sus especificaciones, compilando o interpretando; y generar las clases subyacentes resultantes para su uso por un modelo de objetos en tiempo de ejecución, también según las especificaciones de XAML.

De forma predeterminada, tal procesador interpretará un valor de atributo como una cadena literal o la convertirá en un objeto basándose en el tipo de atributo o en los convertidores de tipos específicos de ese atributo.

2.4 AJAX

AJAX, es el acrónimo de “Asynchronous JavaScript And XML”; es decir JavaScript asíncrono y XML, desarrollada en base a Java Script, para crear aplicaciones interactivas ó RIA (“Rich Internet Applications”).

“Ajax es, una suma de tecnologías ya existentes que facilita la creación de un Internet más fluido, con páginas más ágiles e interactivas, y mayor número de aplicaciones en la Red.”¹¹

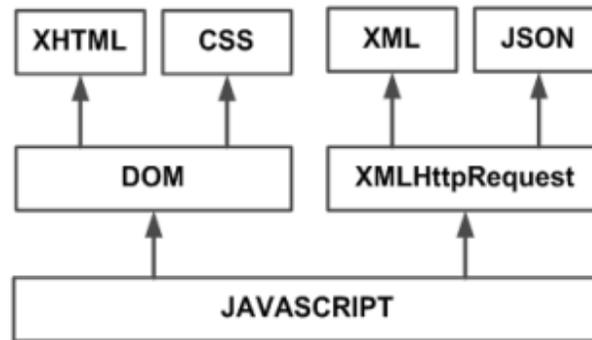
Como tal, AJAX no es una tecnología en sí mismo; sino que se trata de la unión de varias tecnologías que en conjunto conforman una sola tecnología.

¹¹ James Garrett - A New Approach to Web Applications

Estas tecnologías que forman AJAX son:

- **XHTML o HTML y CSS:** Para crear una presentación basada en estándares; es decir hojas de estilos en cascada (CSS) y para el diseño que acompaña a la información.
- **DOM o “Document Object Model ”:** Para la interacción y manipulación dinámica de la presentación; al cual se accede con un lenguaje de “scripting” por parte del usuario, especialmente implementaciones “ECMAScript” como JavaScript y “JScript”, para mostrar e interactuar dinámicamente con la información presentada.
- **XML, XSLT y JSON:** Para el intercambio y manipulación de información.
- **XMLHttpRequest:** empleado para el intercambio asíncrono de información con el servidor Web.
- **JavaScript:** para unir todas las demás tecnologías.

La figura 2.7 expone de una forma más clara la interacción de estas tecnologías, siendo JavaScript la base de AJAX.



Interacción de Tecnologías que componen AJAX

Figura 2.7

El aparecimiento de AJAX como herramienta, propone la implementación de soluciones que requieran resultados en tiempo real, el libro “Introducción a AJAX”¹²; desarrolla a fondo el contenido y funcionalidad de esta herramienta.

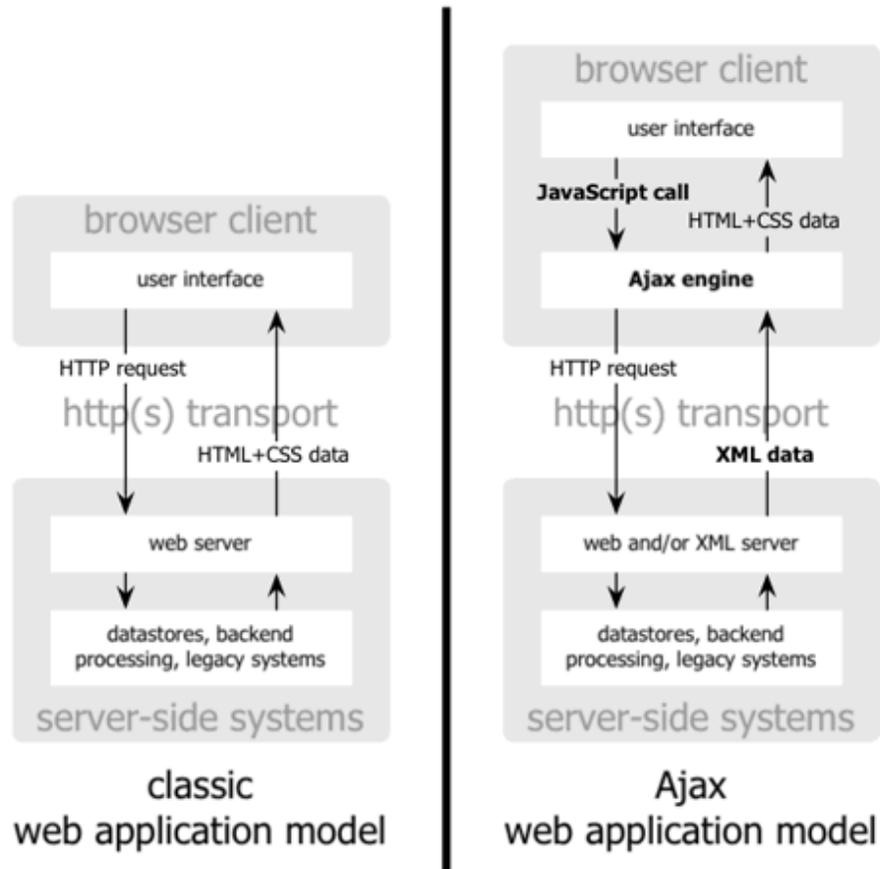
2.4.1 Funcionamiento

El funcionamiento de esta tecnología, se da como respuesta a una tarea que solicita el cliente en el navegador, comunicándose de forma asíncrona con el servidor; trabajando en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla; siendo esto lo que permite aumentar la interactividad, velocidad y usabilidad en la misma.

Como muestra la figura 2.8, en las aplicaciones Web tradicionales las acciones del usuario en la página como presionar un botón, seleccionar un valor de una

¹² Javier Egúiluz Pérez – “Introducción a AJAX” - www.librosweb.es

lista, etc., desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador.



Contraste de aplicaciones Web tradicional con aplicaciones Web con AJAX

Figura 2.8

Esta técnica para crear aplicaciones Web funciona correctamente, pero es muy limitada; y por ende no satisface del todo al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con

los cambios solicitados; si la aplicación debe realizar peticiones continuas, la aplicación Web se convierte en algo más molesto que útil.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Se conoce a AJAX principalmente por su funcionalidad de cargar y “renderizar” una página; para luego mantenerse en esa página, mientras scripts y rutinas van al servidor buscando en “background”, los datos que son usados para actualizar la página solo “re-renderizando” la página y mostrando u ocultando porciones de la misma.

Además suple la deficiencia de respuesta asemejada a “arrancar-frenar-arrancar-frenar”, mediante el motor de AJAX que se encuentra entre el usuario o cliente y el servidor. Aparentemente añadir una capa a la aplicación la haría menos reactiva pero en este caso se produce el efecto opuesto.

En lugar de cargar una página Web al inicio de la sesión, el navegador carga al motor AJAX; el mismo que en la mayoría de casos esta en un frame oculto.

Este motor es el responsable de “renderizar” la interfaz que el usuario ve. Otra actividad que realiza este motor es que permite la interacción asíncrona del

usuario con la aplicación independientemente de la comunicación con el servidor.

Por otro lado; las peticiones HTTP al servidor se transforman en peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor por lo que la respuesta es inmediata.

Si la interacción requiere la respuesta del servidor, la petición se realiza de igual forma asincrónicamente mediante AJAX. En este caso la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta.

2.4.2 Navegadores que soportan AJAX

Debido a que AJAX es una fusión reciente de varias tecnologías; aun existen factores como las características del navegador que determinarán la ejecución efectiva del mismo, pese a esto ha sido un elemento sumamente útil que a ganando fama a nivel general en el entorno de desarrollo, lo cual está rompiendo limitantes de compatibilidad.

A continuación se nombra una lista de navegadores que permiten la ejecución de AJAX.

- Navegadores basados en Gecko: Mozilla, Mozilla Firefox, SeaMonkey, Camino, K-Meleon, Flock, Epiphany, Galeon y Netscape versión 7.1 y superiores.
- Microsoft Internet Explorer para Windows versión 5.0 y superiores; y los navegadores basados en esta plataforma.
- Navegadores con API KHTML versión 3.2 y superiores; incluyendo Konqueror versión 3.2 y superiores.
- Apple Safari versión 1.2 y superiores, y el Web Browser for S60 de Nokia tercera generación y posteriores
- Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores.

Por otro lado; se citan a continuación los navegadores que no soportan contenido AJAX.

- Opera 7 y anteriores.
- Microsoft Internet Explorer para Windows versión 4.0 y anteriores; y versiones para Macintosh, en todas las versiones.
- Dillo.
- Navegadores basados en texto como Lynx y Links.
- Navegadores para incapacitados visuales (braille).

2.5 UML

El Lenguaje Unificado de Modelado conocido como UML, por sus siglas en inglés, “Unified Modeling Language” es el lenguaje empleado en el modelado de sistemas de software; por su concepción y su generalidad es el más conocido y utilizado en la actualidad, que pese a no ser un estándar oficial, está respaldado por el OMG (Object Management Group).

UML se fundamenta en el concepto, de que un modelo es una simplificación de la realidad, que proporciona las características principales de un sistema, permitiendo definir elementos que producen efectos en mayor y menor escala y omitiendo aquellos elementos poco o nada relevantes que requieran un nivel determinado de abstracción.

“El moldeamiento es fundamental en el proceso de desarrollo de software siendo este el referente principal que determinara los lineamientos a seguir durante todas las etapas que dicho proceso contempla.”¹³

UML es un lenguaje gráfico que sirve para visualizar, especificar, construir y documentar un sistema de software, ofreciendo un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema además de aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Es importante resaltar que UML

¹³ Booch, Rumbaugh y Jacobson – “Lenguaje Unificado de Modelado”

es un "lenguaje" para especificar y no para describir métodos o procesos y provee un si un vocabulario y reglas para combinación de palabras a razón de comunicación.

Puede ser aplicado en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, tal como el Proceso Unificado Racional (RUP).

Este lenguaje permite realizar dos tipos de ingeniería que son:

Ingeniería Directa: Permite la creación de código partiendo de un modelo de UML.

Ingeniería Inversa: Permite la creación de un modelo de UML partiendo de la implementación.

En UML un sistema está representado por cinco vistas que permiten describirlo de distintas perspectivas, estas vistas son representadas a través de un conjunto de diagramas que son:

Vista del Usuario: Interpreta al sistema desde el punto de vista de la perspectiva del usuario o actor. Los casos de uso modelan esta vista.

Vista Estructural: Es la estructura estática del sistema en su representación conceptual. (Clases, objetos y relaciones).

Vista del Comportamiento: Representa los aspectos de comportamiento del sistema a través del tiempo. Ejemplo: Diagrama de estado.

Vista de Implementación: Se representan los aspectos estructurales y de comportamiento tal como van a ser implementados.

Vista de Entorno: Aspectos estructurales y de comportamiento en el que el sistema a implementar se representa.

2.5.1 Importancia del Modelamiento

Un lenguaje de modelado es aquel cuyo vocabulario y reglas se enfocan en la representación de la parte conceptual y física de un sistema.

El modelado en el proceso de desarrollo de software, como lo describen Rumbaugh y Jacobson, puede ir desde diseños sencillos que no requieren mayor esfuerzo hasta el moldeamiento de aquellos elementos que requieren una planificación más elaborada que considera factores más complejos como documentos técnicos, estudios de viabilidad, planificación estructural, funcionalidad, seguridad; entre otros.

Los procesos de análisis y diseño, son marcadamente diferentes para el caso de un sistema sencillo, en contraste con un sistema complejo. Por lo tanto, las condiciones de falla o éxito en cada caso producen diferentes niveles de impacto.

Cada sistema puede ser descrito desde diferentes aspectos o empleando diferentes modelos, y cada modelo contiene la abstracción semántica del

sistema. Este modelo puede ser estructurado poniendo énfasis en la organización del sistema, o a su vez; puede enfocarse en la dinámica del sistema analizando su comportamiento.

“Una exitosa organización productora de software es aquella que de manera consistente desarrolla software de calidad enfocado a los requerimientos del usuario tomando en cuenta sus necesidades y su proyección a futuro.”¹⁴

Existe un importante valor en esta afirmación, considerando que el producto primario de un equipo de desarrollo no es la atractiva documentación, tampoco una simple concepción o diagramación de clases, ni los grandes slogans o grandes reconocimientos por un código fuente bien elaborado. Sino un software que satisfaga las necesidades globales de los usuarios u organizaciones; y el rol que dichos sistemas desempeñen en sus negocios, por tanto todos los elementos poco relevantes que giran en torno al proceso de desarrollo de software pasan a un segundo plano. Desafortunadamente muchas organizaciones confunden lo secundario con lo irrelevante.

Para desarrollar un producto de software que satisfaga su propósito primario es necesario lograr que los usuarios expongan de manera concisa y real los verdaderos requerimientos del sistema; además es necesario considerar que para desarrollar software de óptima calidad será necesario construir un fundamento arquitectónico sólido que marque cambios definitivos pero a la vez escalables. De esta forma se logrará implementar los aplicativos requeríos,

14 IB.

previo estudio; de forma rápida, eficiente y efectiva con menores complicaciones.

Para esto se requiere hacer un estudio consistente y en lo posible predecible; con una apreciación objetiva del sistema; incluso considerando su costo en tiempo de vida.

La importancia de los sistemas radica en que es necesario considerar la construcción de modelos que muestren el comportamiento deseado del sistema a elaborarse; además de modelos que permitan visualizar y controlar la arquitectura y aun riesgos que afectan al sistema.

Tomando en cuenta que existen limitantes en la habilidad humana para entender la complejidad, el modelado se puede suplir esta deficiencia ya que se puede alcanzar mayores niveles de abstracción; considerando que los modelos son el reflejo del sistema que se construye.

2.5.2 Diagramas de UML

2.5.2.1. Diagrama de casos de uso

Son diagramas que definen el comportamiento y la funcionalidad de un sistema a construir, detallan la interacción del usuario con el sistema de una forma global pero detallada.

El modelamiento de casos de uso, es la técnica para capturar el comportamiento que se desea del sistema, la misma que permite identificar quién o qué interactúa con el sistema y que debe hacer. Además es una manera de verificar que todos los requerimientos se han levantado.

Los casos de uso son una secuencia de interacciones entre un sistema y alguien que utiliza sus servicios (personas, dispositivos u otros sistemas). Y como tal, debe dejar al sistema en una condición estable y no en medio de un proceso enfocando el ¿Qué? y no el ¿Cómo?

Para emplear este tipo de diagramas; los casos de uso deben ser expresados desde el punto de vista del actor; y deben describir tanto lo que él hace como qué hace el sistema cuando interactúan entre sí.

Elementos de diagramación de Casos de uso

- **Actores**

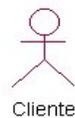


Figura 2.9

- **Caso de uso:**



Figura 2.10

Relaciones de diagramación de Casos de uso

Relaciones «include».-



Figura 2.11

Relaciones «extend».-

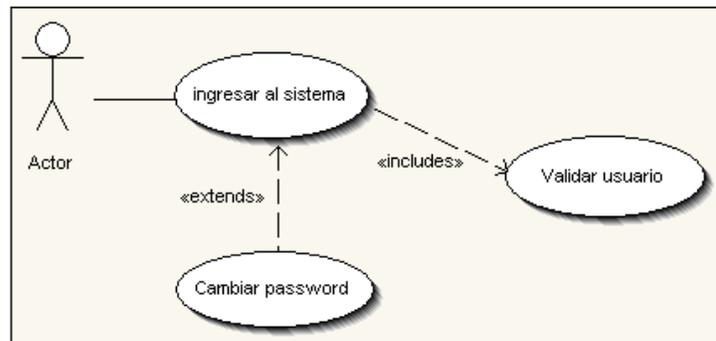


Figura 2.12

Relaciones de herencia.-

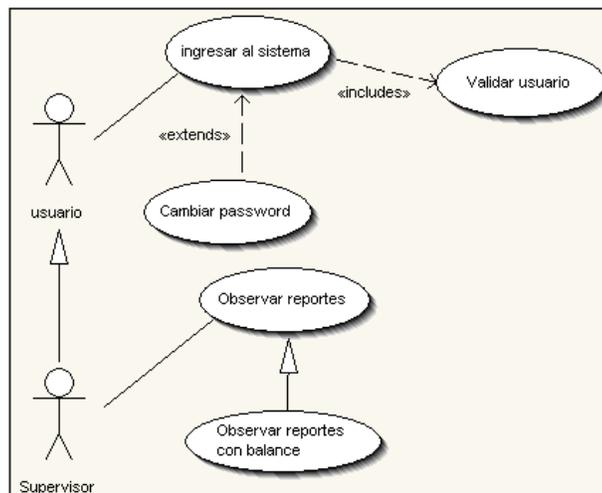


Figura 2.13

Relaciones de dependencia.-

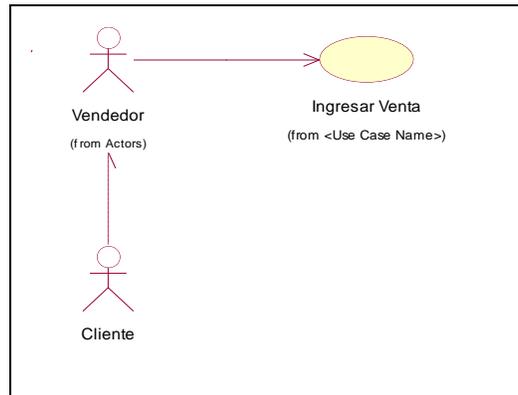


Figura 2.14

2.5.2.2. Diagrama de Clases

Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, objetos, atributos y las relaciones entre ellos. Son utilizados durante el proceso de análisis y diseño conceptual de la información. Su objetivo es representar las estructuras donde se almacenarán los datos, el comportamiento de cada objeto del sistema y sus relaciones con el resto de los objetos.

Elementos de diagramación de Clases.-

- **Atributos**
- **Operaciones**
- **Propiedades**
- **Clases**

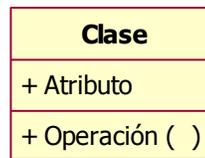


Figura 2.15

Las clases se categorizan a las clases en tres tipos de estereotipos

- Clase control
- Clase entidad
- Clase interfaz

Relaciones de diagramación de Clases

Se definen los siguientes tipos de relaciones

- Agregación
- Composición
- Herencia
- Dependencia



Figura 2.16

2.5.2.3. Diagrama de Componentes

Representan la separación de un sistema de software en componentes físicos (por ejemplo archivos, cabeceras, módulos, paquetes, etc.) y muestra las dependencias entre estos componentes. Estos diagramas proporcionan una

vista física detallada de la organización de los componentes del software, sus interfaces y las dependencias entre ellos. Además de su interacción dentro de un determinado sistema a construirse.

Su objetivo es describir la responsabilidad que se va a tener para conceder la funcionalidad requerida al usuario final. Los elementos de estos modelos son los componentes de software y la dependencia entre ellos. Un componente que puede ser código fuente, código binario, un ejecutable o una librería.

Elementos de diagramación de Componentes

- **Componente**

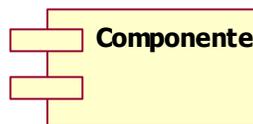


Figura 2.17

- **Interfaz**



Figura 2.18

- **Paquete**

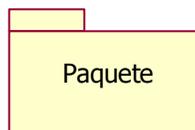


Figura 2.19

2.5.2.4. Diagrama de Despliegue

Se utilizan para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones. En el UML 2.0 los componentes ya no están dentro de nodos. En cambio, puede haber artefactos o algunos nodos dentro de otro nodo.

Un artefacto puede ser algo como un archivo, un programa, una biblioteca, o una base de datos construida o modificada en un proyecto. Estos artefactos implementan colecciones de componentes. Los nodos internos indican ambientes, un concepto más amplio que el hardware propiamente dicho, ya que un ambiente puede incluir al lenguaje de programación, un sistema operativo, un ordenador o un cluster de terminales.

Elementos de diagramación de Componentes

Se representan dos tipos de elementos los cuales son:

- **Nodos**

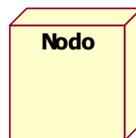


Figura 2.20

- **Conexiones**

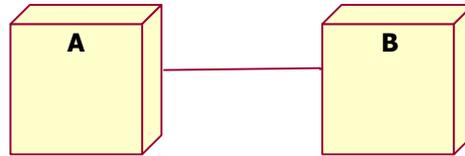


Figura 2.21

2.5.2.5. Diagramas de Secuencia

Es uno de los diagramas más efectivos para modelar la interacción entre objetos en un sistema, a través del tiempo y se modela para cada caso de uso.

Mientras que el diagrama de casos de uso permite el modelado de una vista de acuerdo a la lógica del negocio; el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementarlo; y mensajes pasados entre los objetos.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales.

Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria. Durante el análisis inicial, se coloca el nombre de un mensaje en la

línea del mensaje. Más tarde, durante el diseño, es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado, o invocado, pertenece a la definición de la clase instanciada por el objeto en la recepción final del mensaje.

Elementos de diagramación de Componentes

- Objeto

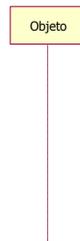


Figura 2.22

Foco de Control

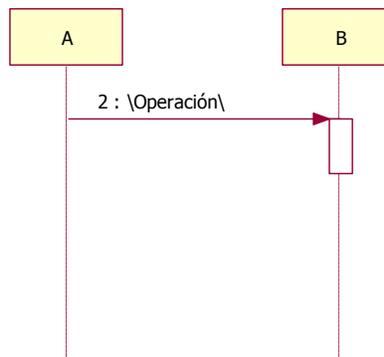


Figura 2.23

2.5.2.6. Diagramas de Colaboración

Esencialmente es un diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los de colaboración muestran explícitamente las relaciones de los roles.

Por otra parte, un diagrama de colaboración no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números tanto la secuencia de mensajes como los hilos concurrentes.

Muestra cómo las instancias específicas de las clases trabajan juntas para conseguir un objetivo común.

Implementa las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro. Dicha implementación es llamada "enlace".

Un diagrama de colaboración es también un diagrama de clases que contiene roles de clasificador y de asociación en lugar de sólo clasificadores y asociaciones. Estos roles describen la configuración de los objetos y de los enlaces que pueden ocurrir cuando se ejecuta una instancia de la colaboración.

Cuando se instancia una colaboración, los objetos están ligados a los roles de clasificador y los enlaces a los roles de asociación. El rol de asociación puede

ser desempeñado por varios tipos de enlaces temporales, tales como argumentos de procedimiento o variables locales. Los símbolos de enlace pueden llevar estereotipos para indicar enlaces temporales.

Elementos de diagramación de Componentes

- **Objeto**

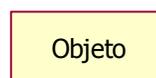


Figura 2.24

- **Vínculo**

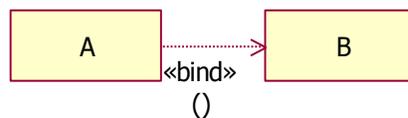


Figura 2.25

2.5.2.7. Diagramas de Actividad

Un diagrama de actividad representa los flujos de trabajo paso a paso de la lógica de negocio y funciones operacionales de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general.

En UML, un diagrama de Actividades es una variación del Diagrama de estados UML donde los "estados" representan operaciones, y las transiciones representan las actividades que ocurren cuando la operación es completa.

Los diagramas de actividad contienen:

- Estados de actividad y de acción.
- Objetos
- Transiciones.

Elementos de diagramación de Actividad

- **Actividad**

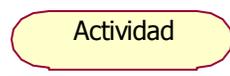


Figura 2.26

- **Inicio**



Figura 2.27

- **Transición**



Figura 2.28

- **Estado**

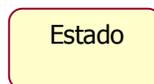


Figura 2.29

- Estado inicial



Figura 2.30

- Estado final



Figura 2.31

2.5.2.8. Diagramas de Estados

Los diagramas de estados muestran el comportamiento de un objeto, es decir, el conjunto de estados por los cuales pasa un objeto durante su ciclo de vida, juntamente con los cambios que permiten pasar de un estado a otro.

Un estado identifica un período de tiempo (no instantáneo) en la vida del objeto, durante el cual está esperando alguna operación; tiene cierto comportamiento característico o puede recibir cierto tipo de estímulos.

Los diagramas de estado resultan adecuados para describir el comportamiento de un objeto a través de diferentes casos de uso, sin embargo, no resultan del todo adecuados para describir el comportamiento que incluye a una serie de objetos colaborando entre sí.

Por lo tanto, resulta útil combinar los diagramas de estado con otras técnicas como lo son los diagramas de interacción; idóneos para la descripción del comportamiento de varios objetos en un único caso de uso, y los diagramas de actividades muestran de forma adecuada la secuencia general de acciones en diferentes objetos y casos de uso.

Elementos de diagramación de Estados

En notación UML, un estado se representa mediante un rectángulo con los bordes redondeados, que se subdividen en:

- Nombre.
- Valor característico de los atributos del objeto en ese estado.
- Acciones que se realizan al entrar, salir o estar en un estado (entry, exit o do, respectivamente).

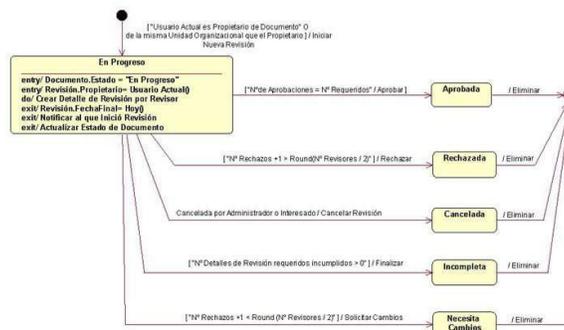


Figura 2.32

La información detallada referente a UML; se encuentra descrita con mayor detalle en el libro de Booch, Rumbaugh y Jacobson bajo el título de “Lenguaje Unificado de Modelado”¹⁵

2.6 RUP

Gestionar los requisitos de un sistema es fundamental en el proceso de desarrollo ya que dependiendo del tamaño del sistema, este puede alcanzar costos muy altos y solo empleando una metodología efectiva este gasto será justificado.

El Proceso Unificado Racional conocido en inglés como “Rational Unified Process”, habitualmente resumido como RUP es un proceso de desarrollo de software que funciona apoyándose en el Lenguaje Unificado de Modelado UML.

Con esta metodología se puede alcanzar un nivel detallado de recolección de datos y manejo de información gracias a los artefactos nativos de la metodología y además porque al ser RUP iterativo, permite abordar en primer lugar las tareas más riesgosas y complejas, dando lugar a una priorización efectiva de requerimientos, dando mayor énfasis a los elementos más importantes.

¹⁵ IB.

2.6.1 Características

RUP es un proceso iterativo; y como tal se adapta a nuevos requisitos o a cambios tácticos en los objetivos del negocio; porque no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

RUP, permite la identificación temprana de riesgos y todas las actividades de desarrollo que se implementan mediante este proceso y están dirigidas por los casos de uso, esto se debe a que el desarrollo de software empleando RUP; se centra en la arquitectura. RUP pone un gran énfasis en la construcción de sistemas basándose en una amplia comprensión de cómo se utilizará el sistema cuando se entregue; a su vez, soporta las técnicas orientadas a objetos.

Los modelos del RUP se basan en los conceptos de “objeto” y “clase”; haciendo de este un proceso configurable que impulsa un control de calidad y una gestión del riesgo objetivos y continuos.

La metodología RUP es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no sea posible cubrir los costos de dedicación del equipo de profesionales necesarios.

2.6.2 Principios de desarrollo

El Proceso unificado Rational; está basado en 6 principios claves:

Adaptar el proceso.- A las características propias del proyecto u organización; el tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

Balancear prioridades.- Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.

Demostrar valor iterativamente.- Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas, en cada una de ellas se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.

Elevar el nivel de abstracción.- Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o esquemas (frameworks) por nombrar algunos.

Un nivel alto de abstracción también permite un análisis completo sobre diversos niveles arquitectónicos, permitiendo adjuntar representaciones visuales de estos, según los diagramas que propone UML.

Enfocarse en la calidad.- El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción

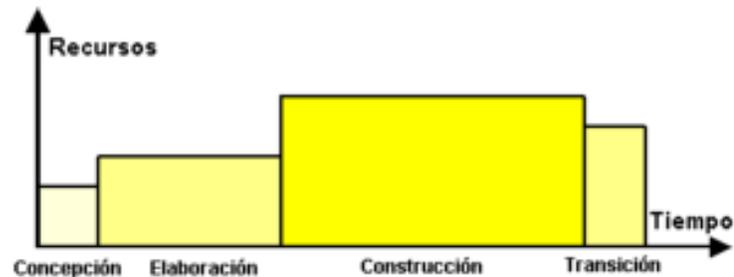
2.6.3 Ciclo de vida

La metodología RUP comprende un ciclo de vida iterativo que organiza tareas en base a fases. Se crea ensamblando los elementos en secuencias semi - ordenadas.

RUP al ser un proceso iterativo al final de cada ciclo, devuelve un resultado determinado. A su vez cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante.

Fases e Iteraciones

La metodología RUP, divide en cuatro fases el desarrollo del software:



Fases de desarrollo de Software

Figura 2.33

- 1. Iniciación o Concepción.-** Es la etapa en la cual se establece la planificación del proyecto (criterios de éxito, evaluación del riesgo, estimación de recursos). El objetivo en esta etapa es determinar la visión y alcance del proyecto; y consiste en hacer un plan de fases, donde se identifican los principales casos de uso y se identifican los riesgos. Además se crea un prototipo ejecutable que sirva para probar los conceptos; para finalmente examinar los objetivos del ciclo de vida del proyecto. Logrando definir si se debe proceder con el desarrollo del sistema.

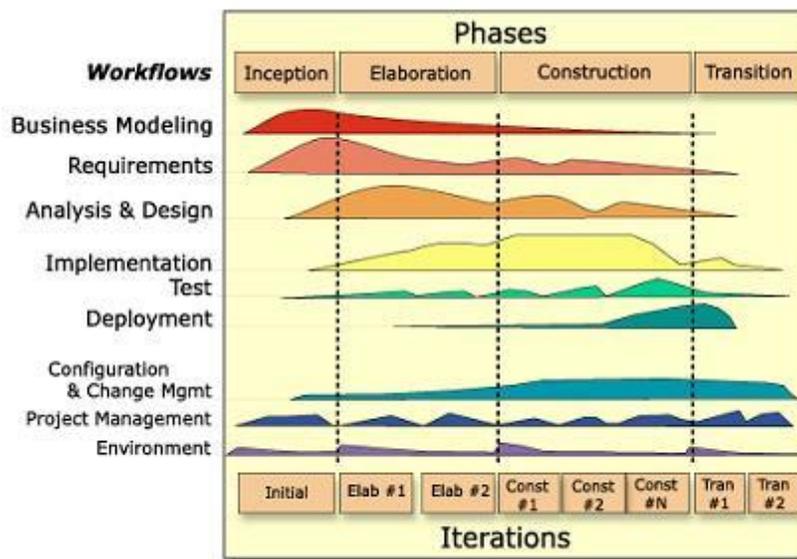
- 2. Elaboración.-** En esta etapa se busca establecer un plan de acción para el proyecto a fin de escoger la arquitectura correcta. Para esto se elabora un plan de proyecto, se completan los casos de uso y se eliminan los riesgos; se analiza el dominio del problema; y se establece una base arquitectónica sólida; para finalmente decidir si pasar o no a la etapa de construcción.

- 3. Construcción.-** Etapa en la cual, el objetivo es llevar a obtener la capacidad operacional inicial, desarrollando un producto totalmente operativo y eficiente incluyendo el manual de usuario.

En esta etapa, se desarrolla de forma iterativa e incremental un producto completo. Se describen los requisitos restantes y se establecen los criterios de aceptación. También se redefine el diseño y se completa la implementación, para finalmente ejecutar las pruebas del software.

- 4. Transición.-** Para esta etapa se busca entregar el sistema a los usuarios finales, esta contempla actividades como, instalación el producto para el cliente y entrenamiento a los usuarios; como consecuencia de esto suelen surgir nuevos requisitos, por lo tanto, se analizan y corrigen los problemas no detectados. Además; se finalizan las características antes pospuestas y al final de la fase se decide si los objetivos se han cumplido; determinando si se debe empezar otro ciclo de desarrollo.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.



Fases e Iteraciones de la Metodología RUP
Figura 2.34

La figura 2.34 representa la arquitectura del RUP, que posee dos dimensiones, las cuales representan en el eje horizontal el tiempo, y presenta los aspectos del ciclo de vida de los procesos y como estos son desplegados. Esta primera dimensión ilustra el aspecto dinámico de los procesos expresados en términos de fases, iteraciones e hitos.

El eje vertical representa las disciplinas que agrupan actividades. Esta segunda dimensión presenta el aspecto estático de los procesos, descrita en términos de componentes de procesos, disciplinas, actividades, workflows, artefactos y roles.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada iteración.

Cabe mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- **Ingeniería de Negocios:** Entendiendo las necesidades del negocio.
- **Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.

- **Análisis y Diseño:** Trasladando los requerimientos dentro de la arquitectura de software.
- **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado este presente.

Disciplina de Soporte

- **Configuración y administración del cambio:** Guardando todas las versiones del proyecto.
- **Administrando el proyecto:** Administrando horarios y recursos.
- **Ambiente:** Administrando el ambiente de desarrollo.
- **Distribución:** Hacer todo lo necesario para la salida del proyecto

Ventajas del desarrollo iterativo.- Un desarrollo iterativo es generalmente superior a un desarrollo tradicional debido a que, los riesgos son disminuidos con mayor facilidad, ya que los elementos son integrados progresivamente; permitiendo además el ajuste de cambios en requerimientos y tácticas para que:

- Las organizaciones puedan aprender de este acercamiento e improvisar sus procesos.
- La reusabilidad sea incrementada.

Manejo de Requerimientos.- Es un sistemático acercamiento para descubrir, documentar, organizar y rastrear los cambios en los requerimientos del sistema.

La clave del efectivo manejo de requerimientos incluye mantener una clara definición de estos, con apropiados atributos y trazabilidad a otros requerimientos y artefactos de proyectos.

Entre las principales dificultades para encontrar requerimientos se tiene:

- No son siempre obvios.
- No siempre son fáciles y claros de expresar.
- Hay diferentes niveles de requerimientos y diferentes niveles de detalle.
- Cambian y estos impactan en otros requerimientos.

2.6.4 Elementos de RUP

Se clasifican de la siguiente forma:

Actividades.- Son los procesos que se determinan en cada iteración.

Trabajadores o Desarrolladores.- Son las personas o entes involucrados en cada proceso.

Artefactos.- Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración se hace exigente el uso de artefactos, siendo por este motivo una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

2.6.5 Artefactos de RUP

Son los informes documentados o a su vez archivos ejecutables que se producen, manejan o ejecutan. Un artefacto es el producto o subproducto de un proceso determinado. Los artefactos pueden aparecer como elementos de entrada o salida.

Entre los principales artefactos que se emplean en la metodología se pueden mencionar:

Documento de visión.- Define a los afectados del proyecto, sus necesidades y la relación con el producto a ser desarrollado, especificando además las características del mismo.

El documento de visión comunica la información entre la dirección, el control de comercialización y el equipo del proyecto, describe “el qué” “y por qué” del producto. Se establece el alcance y la prioridad de peticiones.

Debido a que este documento es analizado por varios miembros del personal implicado, el mismo deberá ser detallado a un nivel general para que se lo

pueda entender, por lo tanto debe proporcionar la información requerida para crear el modelo de casos de uso.

Documento de especificaciones suplementarias.- Este documento se lo realiza en las fases de inicio y elaboración, en este se describen los requisitos funcionales y los no funcionales encontrados al realizar el modelo de casos de uso. Este artefacto puede reunir los requerimientos del sistema que no fueron reunidos en el modelo de casos de uso. El documento de especificaciones suplementarias contiene los requerimientos no funcionales del sistema.

Documento de especificación de casos de uso.- Muestra la funcionalidad de cada caso de uso del sistema, estos contienen los requerimientos detallados del software. La descripción de los mismos se efectúa para construir el contorno del flujo de acontecimientos, sin esta descripción es casi imposible desarrollar una solución.

El documento de requerimientos es para que los usuarios los lean y lo entiendan, por lo cual es importante que sea escrito de forma adecuada.

La especificación de casos de uso describe detalladamente un caso de uso específico. Al mismo tiempo del flujo de acontecimientos, existen adicionalmente otras propiedades que lo describen.

Se recomienda usar la plantilla de RUP para describir los casos de uso del sistema. El flujo de acontecimientos contiene las acciones del actor y lo que el sistema da en respuesta a esas acciones.

Las funciones que cumple el sistema son los requerimientos del software, es decir, lo que el usuario quiere que el sistema haga.

Sub Flujos.- Pueden ocasionar un incremento en el entendimiento y permite la reutilización interna de los requerimientos. Un flujo llama a un Sub flujo; cuando este completa su acción siempre vuelve a la línea después de la que le llamaron; es similar al concepto de llamada en algunos lenguajes de programación. Como cualquier flujo, un Sub flujo puede tener flujos alternos.

Flujos Alternos.- Un flujo alternativo puede ocurrir dentro de otro flujo. Estos tienen la ventaja de hacer que un flujo sea fácil de entender para ponerlos en práctica. Cuando se usan se deben identificar sus escenarios. Cuando se pone en práctica un caso de uso, se lo hace en varios escenarios a la vez, es por esto que se debe asegurar que todos los flujos alternos estén incluidos en al menos un escenario a fin de que sea puesto en práctica.

Precondiciones.- Una precondición para un caso de uso es una condición en el sistema que tiene que ser verdadera antes de que empiece el caso de uso;

una precondition no es el acontecimiento que hace que comience el caso de uso. Estas condiciones son opcionales.

Las precondiciones disminuyen la necesidad de validación dentro de los flujos de los casos de uso. Estas no describen cosas que estén fuera del sistema.

Pos condiciones.- Es una condición luego de que finalice el caso de uso, es garantizada para ser verdadera; tiene que ser positivo para un caso de uso independientemente de cuáles flujos alternativos fueron ejecutados. Las pos condiciones son un instrumento eficaz para desarrollar casos de uso; pueden ser utilizadas al final de los flujos alternativos o sub flujos.

2.6.6 Modelos

Son el tipo de artefacto más importante en el RUP. Es una simplificación de la realidad, creada para comprender mejor el sistema. RUP presenta los nueve modelos más importantes:

Modelo del negocio.- Establece una abstracción de la organización.

Modelo del dominio.- Establece el contexto del sistema.

Modelo de casos de uso.- Establece los requisitos funcionales del sistema.

Modelo de análisis.- Establece un diseño de las ideas.

Modelo de diseño.- Establece un vocabulario del problema y su solución.

Modelo del proceso.- Establece los mecanismos de concurrencia y sincronización del sistema.

Modelo de despliegue.- Establece la topología de hardware sobre la cual se ejecutará el sistema.

Modelo de implementación.- Establece las partes que se utilizarán para ensamblar y hacer disponible el sistema físico.

Modelo de pruebas.- Establece las formas de validar y verificar el sistema.

RUP considera importante que se evalúe la calidad de todos los artefactos en varios puntos en el ciclo de vida del proyecto; y se recomienda que se lo haga antes de concluir cada iteración.

Durante la producción del software ejecutable, cada iteración debe someterse a la demostración y prueba para eliminar defectos arquitectónicos.

CAPÍTULO III

ANALISIS DEL SISTEMA

3.1 Descripción del Problema

La empresa Donaire Needle Point, creada en el año 1992. Es una empresa textil productora de uniformes, especializada en línea deportiva. Que en la actualidad se encuentra automatizando sistemas y procesos que agilicen la producción y comercialización de sus productos, en sus diversas categorías.

La necesidad de un sistema de cotización de uniformes, se fundamenta en que a pesar de los avances tecnológicos y los lanzamientos de software para diversas disciplinas comerciales y funciones administrativas; no existe un software específico desarrollado para la cotización de uniformes que contemple los principales factores que influyen en el costo de la elaboración de prendas.

Todas las cotizaciones para la elaboración de uniformes; requieren de la presencia del personal de ventas de la empresa, creando una dependencia, incluso para la promoción mínima de productos.

Debido a que en la actualidad existe una difusión y una promoción comercial masiva mediante los medios de comunicación incluyendo al Internet, la existencia de un sitio Web empresarial se convierte en un elemento necesario para la promoción de la empresa como tal.

El presente sistema tiene como objetivo suplir este conjunto de requerimientos; mediante un sistema integrado en un catálogo en línea empleando herramientas visuales y funcionales de última generación, que permitan al usuario definir su pedido en el proceso de compra; y a la empresa Donaire Needle Point; la facilidad de exponer sus productos y servicios las 24 horas los 365 días del año.

3.2 Documento de Visión

VISIÓN DEL SISTEMA

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

Versión 1.2.0

Visión del Sistema

Información General

Título: "Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML"

Versión: 1.2.0

Estado: Documento Aprobado

Firmas y Aprobaciones

Elaborado Por: Juan David Utreras Jácome

Cargo Postulante al título de Ingeniero de Sistemas

Fecha: Noviembre 2008

Firma: _____

Revisado Por: Ing. Santiago Albuja

Cargo Profesor Guía

Fecha: Noviembre 2008

Firma: _____

Aprobado Por: Dra. Olga Jácome

Cargo Propietaria de Donaire Needle Point

FECHA: Noviembre 2008

Firma: _____

Lista de Cambios

Versión	Fecha	Autor	Descripción
1.0.0	Mayo /08	JDUJ	Eliminación de redundancia de datos y tablas en Base de Datos del sistema.
1.0.0	Julio/08	JDUJ	Cambio en documento de Visión
1.1.0	Agosto/08	JDUJ	Ajuste en Sub división de Módulos
1.2.0	Octubre/08	JDUJ	Inserción de interfaces para Parametrización

3.2.1 Introducción

Ante la necesidad de la Empresa Donaire Needle Point de contar con un catálogo en línea que le permita promocionar sus productos, y captar clientes mediante una ágil forma de cotizar, al mismo tiempo que posiciona una imagen corporativa, se ha considerado incluir un subsistema capaz de generar cotizaciones de uniformes de acuerdo a parámetros escogidos por el cliente; tomando en cuenta la funcionalidad de tecnologías de última generación.

3.2.2 Propósito

Especificar las características del sistema de cotización de uniformes para la empresa Donaire Needle Point, el mismo que permitirá exponer la línea de productos, las alternativas de modificación en las prendas predefinidas y al mismo tiempo permitirá agilizar la generación de cotizaciones.

3.2.3 Alcance

El sistema de cotización de uniformes mediante catálogo en línea, cumple con la siguiente funcionalidad: introducción y refuerzo de imagen corporativa de Donaire Needle Point; promoción y exhibición de productos de la empresa, por categorías; generación de cotizaciones generales y específicas; control de las proformas generadas.

Para el cumplimiento de su objetivo, se desarrollan las siguientes etapas:

- Investigación de las tecnologías WPF, Silverlight, XAML, AJAX y ASP.NET

- Uso de la metodología RUP y UML para Análisis y diseño de la aplicación.
- Segmentación de módulos del sistema
- Concepción gráfica del sistema orientada a Interfaces de Usuario
- Implementación de Base de Datos de sistema
- Creación de entorno multicapas para aplicación Web
- Construcción de modelo base de sistema
- Construcción de modelos específicos de sistema
- Ejecución de pruebas y retoques de sistema

De manera preliminar, se han identificado tanto a los usuarios como a los afectados del mismo mediante el análisis de los se relacionan con el sistema y que se detallan en el presente documento.

3.2.4 Posicionamiento del Producto

Definición del Problema

El problema de	<ul style="list-style-type: none"> • Carencia sitio Web que exponga los productos de la empresa. • Carencia de sistematización en cotización de uniformes, que genera una dependencia de "persona a persona". • Insuficiencia en registro de clientes actuales y potenciales. • Inexistencia de imágenes de productos de la empresa • Inexistencia de segmentación de categorías de productos • Deficiencia en el ingreso, modificación, inserción y eliminación de productos
Afecta a	<ul style="list-style-type: none"> • Clientes. • Personal de Ventas. • Personal Administrativo. • Accionistas de la Empresa. • Empleados Manufactureros. • Proveedores de materia prima.
Cuyo impacto es	<ul style="list-style-type: none"> • La falta de seguimiento a clientes reales y captación a clientes potenciales. • Que los clientes no dispongan oportunamente de información de precios y alternativas de productos. • La dependencia de la presencia del personal de ventas. • La deficiencia en el cálculo estimado de precios de productos. • La falta de difusión de los productos de la empresa. • La sobre inversión de tiempo del personal de la empresa. • La deficiencia de administración que afecta al costo beneficio de la empresa.

Una solución exitosa	<ul style="list-style-type: none"> • Será el registro de clientes reales y potenciales que visitan la página Web. • Implementará un aplicativo Web en el que se exhiba los productos de la empresa con sus precios y organizados de acuerdo a sus categorías; dando al cliente una amplia visión de las líneas optativas para su necesidad particular. • Agilitará el Proceso de Cotización de prendas, evitando en lo posible la presencia del personal de ventas. • Generará estandarización en precios y categorías de productos. • Permitirá un mayor alcance Promocional mediante el uso de herramientas Web promocionando la nueva imagen corporativa de la empresa. • Optimizará los recursos de la empresa incrementando el costo beneficio.
----------------------	--

Posicionamiento del Producto

Para	Empresa "Donaire Needle Point"
Quién	<ul style="list-style-type: none"> • Necesita un catálogo en línea que promocióne sus productos, manteniendo la información organizada según categorías; tomando en cuenta las alternativas de cada producto y sus características de composición. • Requiere que el aplicativo Web promocióne y refuerce la imagen corporativa de la empresa, de una manera vistosa y útil.

	<ul style="list-style-type: none"> • Necesita un sistema efectivo que permita generar cotizaciones en tiempo real previo registro del cliente. • Necesita un sistema que permita tener una buena administración del contenido de sus páginas Web.
El Sistema de Cotización de Uniformes mediante catálogo en línea	<ul style="list-style-type: none"> • Es un sistema integrado dentro de un sitio Web, empleando técnicas visuales atractivas y útiles mediante el uso de herramientas como "Microsoft Silverlight", AJAX y ASP.NET.
Que	<ul style="list-style-type: none"> • Da a los clientes una amplia información y alternativas de los productos de la empresa. Información disponible desde cualquier lugar del mundo a través del Internet, mediante un navegador. • Ofrece a los clientes la posibilidad de obtener cotizaciones inmediatas de acuerdo a parámetros seleccionados ese momento. • Permite actualizar y administrar el contenido del sitio Web oficial de la empresa Donaire Needle Point. • Permite que los clientes registren sus datos al momento de efectuar su cotización; re-direccionando esta información de forma confidencial a la base de datos de la empresa. • Permite a los clientes obtener una cotización estimada real de acuerdo a las prendas elegidas. • Registra en la base de datos de la empresa las cotizaciones que el cliente efectúa.
A Diferencia	<ul style="list-style-type: none"> • De no tener un sistema de catálogo en línea que promocióne a la empresa y sus productos. • De no contar con un sistema computarizado que permita la cotización estimada inmediata de costos de producción de uniformes.

	<ul style="list-style-type: none"> • De no contar con un sistema automatizado que permita administrar la información relativa a los productos de la empresa. • De no tener un sistema que registre información referente a clientes reales y potenciales.
Esta Aplicación	<ul style="list-style-type: none"> • Informa y posiciona la imagen corporativa de Donaire manteniendo una armonía cromática y una lógica de composición de imagen. • Expone los productos de la empresa agrupándolos por categorías según prendas de tipo edades, genero, tipo, calidad, color, y composición de la prenda. • Cotiza prendas partiendo de un modelo base personalizable de acuerdo a los requerimientos de los usuarios. • Promociona novedades relativas a tendencias de moda y la industria textil. • Administra el contenido de catálogo, parametrizando el uso de la información expuesta y recibida por los clientes.

3.2.5 Descripciones de Afectados y Usuarios

Resumen de los Afectados

Nombre	Descripción	Responsabilidades
Clientes	Personas interesadas en los productos y servicios de la empresa.	<ul style="list-style-type: none"> • Satisfacer su necesidad de compra.
Personal de Ventas.	Personas capacitadas en ventas que trabajan para la empresa.	<ul style="list-style-type: none"> • Promocionar productos de la empresa. • Captar nuevos clientes.
Personal Administrativo	Personas capacitadas en administración que trabajan para la empresa.	<ul style="list-style-type: none"> • Administrar la información referente a clientes reales y potenciales.
Accionistas de la empresa	Personas que invierten recursos para correr la empresa.	<ul style="list-style-type: none"> • Mantener un control permanente del costo beneficio de la empresa.
Empleados Manufactureros	Personal capacitado para la manufactura de productos.	<ul style="list-style-type: none"> • Producir los productos requeridos.
Proveedores de Materia Prima	Fábricas, y distribuidores de insumos para la producción.	<ul style="list-style-type: none"> • Proveer materia prima demanda en el mercado.

Resumen de usuarios

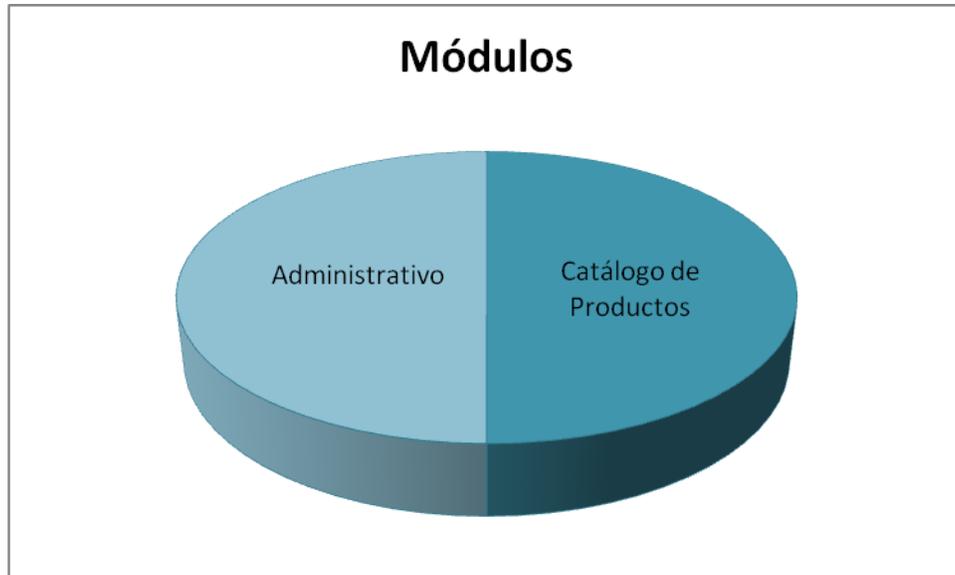
Nombre	Descripción	Afectado al que representa
Clientes	Personas Interesadas en la adquisición de uniformes.	Clientes
Personal Administrativo	Persona encargada de la administración del contenido de las páginas Web.	Empresa
Webmaster	Persona encargada de la administración del contenido de las páginas Web, con acceso directo al sistema.	Empresa

Principales necesidades de los Afectados / Usuarios

Necesidad	Beneficio	Complejidad
Obtener información oportuna sobre las alternativas de productos.	Alto	Media
Contar con un catálogo y listado de precios actualizado.	Alto	Alta
Contar con la información de clientes reales y potenciales	Alto	Alta
Optimización de Costo Beneficio	Alto	Medio
Obtener cotización de uniformes de acuerdo a parámetros definidos por el usuario	Medio	Alta
Administrar el contenido de catálogo en línea	Alto	Media

3.2.6 Resumen del Producto

Perspectiva del producto



Módulos

Figura 3.1

Como lo indica la figura 3.1, el Sistema de Cotización mediante catálogo en línea, constará de dos módulos, que son:

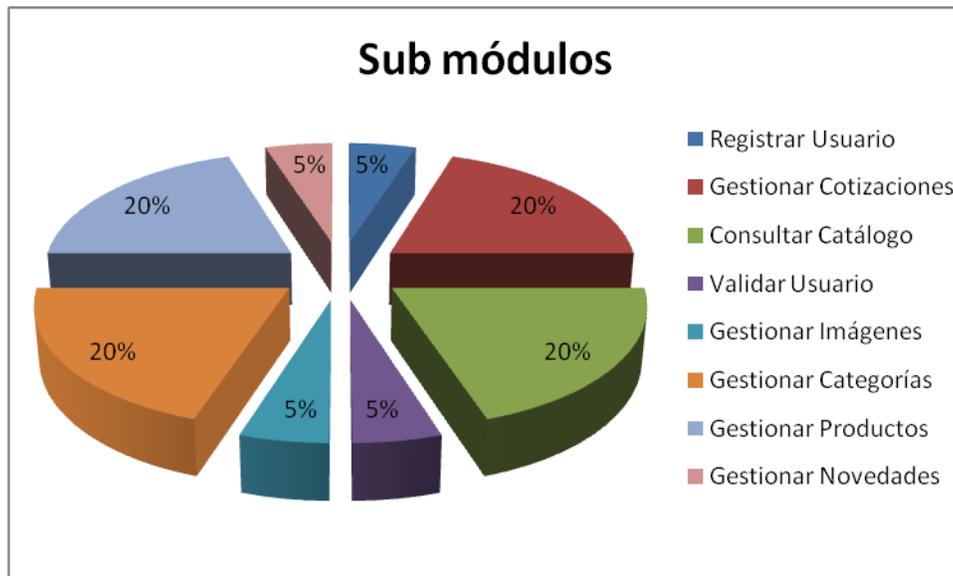
Módulo de Catálogo de Productos.- Se encargará de la administración de todo el contenido del catálogo; como imágenes, información relativa a los productos y líneas determinadas. Este a su vez esta segmentado en los siguientes Sub Módulos:

- Registrar Usuario
- Gestionar Cotizaciones
- Consultar Cotizaciones

- Validar Usuario

Módulo Administrativo.- Permitirá gestionar el contenido de la aplicación de catálogo en línea. Y este también se subdivide en los siguientes sub módulos:

- Gestionar Imágenes
- Gestionar Categorías
- Gestionar Productos
- Gestionar Novedades



Sub Módulos

Figura 3.1

Características del Producto

Módulo - Catálogo de Productos				
Característica	Descripción	Probabilidad Cambio	Beneficio	Prioridad
Registro de Usuario	Almacena la información ingresada por el usuario que desea efectuar cotizaciones.	Alta	Medio	1
Gestión de Cotizaciones	Permite obtener a los usuarios registrados, obtener cotizaciones partiendo de modelos básicos de uniformes los mismos que pueden ser modificados y valorados en el momento solicitado.	Alta	Alto	1
Consultar Catálogo	Despliega de manera organizada mediante categorías, los productos de la empresa	Baja	Alto	1
Validar Usuario	Permite efectuar una validación de usuario, con la finalidad de restringir el acceso a las interfaces protegidas para uso exclusivo de los clientes registrados y el administrador del sitio	Baja	Alto	1

Módulo - Administrativo				
Característica	Descripción	Probabilidad Cambio	Beneficio	Prioridad
Gestionar Imágenes	Permite añadir, eliminar, actualizar y consultar imágenes relativas a los productos.	Baja	Medio	2
Gestionar Categorías	Permite actualizar las categorías disponibles.	Baja	Alto	1
Gestionar Productos	Administra los productos y los elementos que los componen; es decir factores como: color, extras, calidades, precios y toda la información diferencial.	Alto	Alto	1
Gestionar Novedades	Permite ingresar información adicional relativa a tendencias de moda, cromática, noticias, y novedades.	Bajo	Bajo	2

3.3 Resumen Ejecutivo Técnico del Sistema

RESUMEN EJECUTIVO TÉCNICO DEL SISTEMA

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

Versión 1.2.0

Introducción

Este documento tiene como objetivo desarrollar el análisis de los requerimientos para poder realizar los diseños lógicos y físicos, los cuales garanticen una correcta construcción del sistema propuesto.

Objetivos de Diseño del Sistema

- Estructurar los artefactos que permitan desarrollar el sistema basándose en los requerimientos predefinidos.
- Definir el proceso para la aplicación de técnicas y principios con el fin de definir el sistema a implementar con el detalle suficiente para permitir su interpretación y realización.
- Representar gráficamente el funcionamiento del sistema a implementar.

Requerimientos del Negocio

Los requerimientos de Donaire Needle Point son: Promocionar al público en general los productos creados por la empresa; brindando además la funcionalidad a los clientes registrados la alternativa de conocer por sí mismos el costo aproximado de producción de las prendas de su interés. Se considera además la importancia de una interface de usuario atractiva que permita que el contenido y la presentación de productos incentiven al cliente a una decisión final de compra.

Requerimientos de Usuario

Los requerimientos de usuario se clasifican según el tipo de usuario del sistema

Clientes

- Obtener información relativa al catálogo de productos de la empresa
- Enterarse de novedades relativas a moda y diseño.
- Ingresar información.
- Crear cotizaciones personalizadas.

Webmaster

- Administración del contenido del sitio Web de catálogo de productos de Donaire Needle Point.
- Creación de elementos funcionales del sitio Web y de tipo interface.
- Mantenimiento del sitio Web.

Requerimientos Adicionales del Sistema

- Manejo y mantenimiento de la base de datos del sistema que garantice el funcionamiento correcto del mismo.
- Implementar seguridad de validación de usuarios registrados, para reducir la probabilidad de intrusiones no autorizadas en el sistema.

3.4 Especificaciones suplementarias

Las especificaciones suplementarias del sistema se encuentran detalladas en el anexo “A”

3.5 Especificaciones de CASOS DE USO

Se detalla el caso de uso “Gestionar Cotizaciones” pues es el de mayor importancia. Las especificaciones de casos de uso restantes del sistema se encuentran detalladas en el anexo “B”.

Gestionar Cotizaciones

3.5.1 Breve Descripción

Este caso de uso muestra la funcionalidad del sistema en lo referente a la generación de cotizaciones o proformas partiendo de parámetros ingresados por el cliente.

3.5.2 Precondiciones

Existencia de Registro de Cliente

Debe haber por lo menos un registro de usuario tipo cliente en el sistema. Es decir, el usuario cliente debe haber ingresado sus datos personales, y debe haber escogido una contraseña y un nombre de usuario. Los mismos que serán solicitados previo ingreso a Gestión de Cotizaciones.

Haber iniciado sesión como usuario Cliente o Administrador

Es necesario haber iniciado sesión un usuario registrado en el sistema.

Obtención válida de elementos de Catálogo

Debe haberse cargado los datos referentes a catálogo de productos.

3.5.3 Pos condiciones de éxito

Cotización generada exitosamente

Se genera una cotización según los parámetros escogidos por el cliente y se guarda en la base de datos con un número único previa aceptación por parte del usuario.

3.5.4 Pos condiciones de falla

No se pueden mostrar las opciones de catálogo

El sistema no es capaz de desplegar las opciones de catálogo debido a una falla de conectividad.

No se generó la cotización efectuada por el usuario, debido a un error de disponibilidad del sistema

El sistema no es capaz de generar la cotización por fallas internas del sistema en el proceso de despliegue de datos o ejecución efectiva de cálculos referentes a la cotización.

No se registró la cotización efectuada por el usuario, debido a un error en la base de datos del sistema

El sistema no es capaz de almacenar la cotización generada en la base de datos del sistema.

No se registró la cotización efectuada por el usuario; debido a una falla en la red

El usuario no puede registrar la cotización efectuada a causa de una falla en la red.

3.5.5 Actor Principal

Cliente (Usuario)

3.5.6 Flujo de Eventos

3.5.6.1. Flujo Básico (FB)

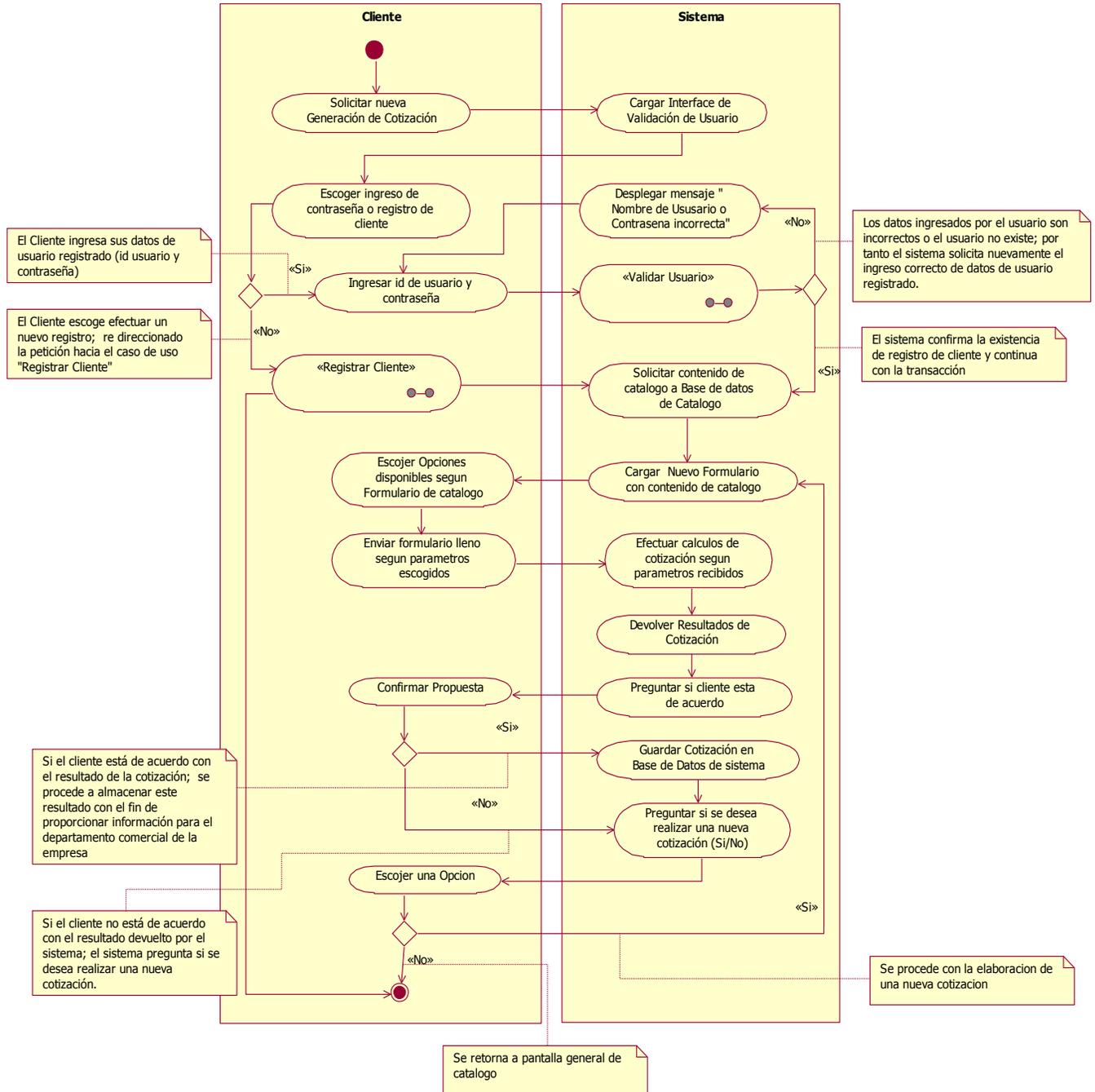
1. El caso de uso inicia cuando el usuario solicita generar cotización de uniformes.
2. El sistema carga la interface de validación de usuario.
 - a. FA¹⁶ El usuario escoge “Registrar Nuevo Usuario”.
 - b. Se direcciona a FB Registrar nuevo Usuario
 - c. Termina Caso de Uso.
3. El usuario Escoge ingresar nombre de usuario y contraseña.
4. El usuario ingresa su “id de usuario” y su contraseña.
 - a. Flujo de Excepción (FE) - El usuario no ingresa sus datos.
 - b. Termina el caso de uso.
5. El sistema invoca a FB Validar usuario y envía parámetros ingresados.
 - a. Flujo Alterno 2 (FA2) - El usuario ingreso incorrectamente su “id de usuario” y/o contraseña.
 - b. Retornar al punto 3 del Flujo Básico.

¹⁶ Flujo Alterno (FA) – Ver ANEXO “B” “Registrar Usuario”

6. El sistema confirma la existencia de usuario según los parámetros ingresados y solicita contenido de catálogo a la base de Datos del sistema.
7. El sistema, carga un nuevo formulario con el contenido del catálogo.
8. El usuario escoge las opciones disponibles según el formulario de catálogo.
9. El usuario envía el formulario lleno según parámetros escogidos.
10. El sistema efectúa el cálculo de la cotización según parámetros recibidos.
11. El sistema devuelve los resultados de la cotización.
12. El sistema consulta al usuario si está de acuerdo con el resultado devuelto.
13. El cliente confirma su respuesta a favor.
 - a. Flujo Alternativo FA3 (FA3) - el cliente confirma su respuesta en desacuerdo con el resultado
 - b. Retornar al punto 7 del flujo básico.
14. El sistema pregunta si el Usuario desea realizar una nueva cotización.
 - a. Flujo Excepción (FE1) - El usuario sale de la aplicación.
 - b. Termina el caso de uso.
15. El usuario escoge salir del módulo de cotización.
 - a. Flujo Alternativo (FA4) - El usuario escoge continuar con una nueva cotización.
 - b. Retornar al punto 7 del flujo básico
16. Termina el caso de uso.

3.5.7 Diagramas

Outline de Gestionar Cotizaciones



Interfaz gráfica con el usuario

Pantalla de Ingreso de Usuario



Pantalla de Parámetros Predefinidos para cotización

Prenda:
 Talla:
 Tela:
 Color Primario:
 Color Secundario:

El costo Aproximado de su Uniforme es: **3.6**

Extras

<input type="checkbox"/> Estampado monocromatico pequeno	<input checked="" type="checkbox"/> Estampado monocromatico mediano	<input type="checkbox"/> Estampado monocromatico grande	<input type="checkbox"/> Estampado multicolor pequeno
<input type="checkbox"/> Estampado multicolor mediano	<input type="checkbox"/> Estampado multicolor grande	<input type="checkbox"/> Estampado tricolor pequeno	<input type="checkbox"/> Estampado tricolor mediano
<input type="checkbox"/> Estampado tricolor grande	<input type="checkbox"/> Bordado monocromatico pequeno	<input type="checkbox"/> Bordado monocromatico mediano	<input type="checkbox"/> Bordado monocromatico grande
<input type="checkbox"/> Bordado multicolor pequeno	<input type="checkbox"/> Bordado multicolor mediano	<input type="checkbox"/> Bordado multicolor grande	<input type="checkbox"/> Botones monocromatico pequeno
<input type="checkbox"/> Botones monocromatico mediano	<input type="checkbox"/> Botones monocromatico grande	<input type="checkbox"/> Cierres monocromatico pequeno	<input type="checkbox"/> Cierres monocromatico mediano
<input type="checkbox"/> Cierres monocromatico grande	<input type="checkbox"/> Cuello multicolor pequeno	<input type="checkbox"/> Cuello multicolor mediano	<input type="checkbox"/> Cuello multicolor grande
<input type="checkbox"/> Tira monocromatico pequeno	<input type="checkbox"/> Tira monocromatico mediano	<input type="checkbox"/> Tira monocromatico grande	<input type="checkbox"/> Capucha monocromatico pequeno
<input type="checkbox"/> Capucha monocromatico mediano	<input type="checkbox"/> Capucha monocromatico grande	<input type="checkbox"/> Capucha bicolor pequeno	<input type="checkbox"/> Capucha bicolor mediano
<input type="checkbox"/> Capucha bicolor grande	<input type="checkbox"/> Broche monocromatico pequeno	<input type="checkbox"/> Broche monocromatico mediano	<input type="checkbox"/> Broche monocromatico grande
<input type="checkbox"/> Puño multicolor pequeno	<input type="checkbox"/> Puño multicolor mediano	<input type="checkbox"/> Puño multicolor grande	<input type="checkbox"/> Cordon monocromatico pequeno
<input type="checkbox"/> Cordon monocromatico mediano	<input type="checkbox"/> Cordon monocromatico grande		

Para la generación de cotizaciones, la interface actual muestra parámetros predefinidos que se cargan conforme el cliente escoge de las alternativas existentes, para obtener el valor individual de cada prenda.

3.5.8 Relaciones

Interfaces con otros sistemas

No existen interfaces con otros sistemas.

Inclusiones

Este caso de uso incluye el caso de uso “Consultar Catálogo”.

Este caso de uso incluye el caso de uso “Validar Usuario”.

Extensiones

Este caso de uso se extiende al caso de uso “Registrar Usuario”.

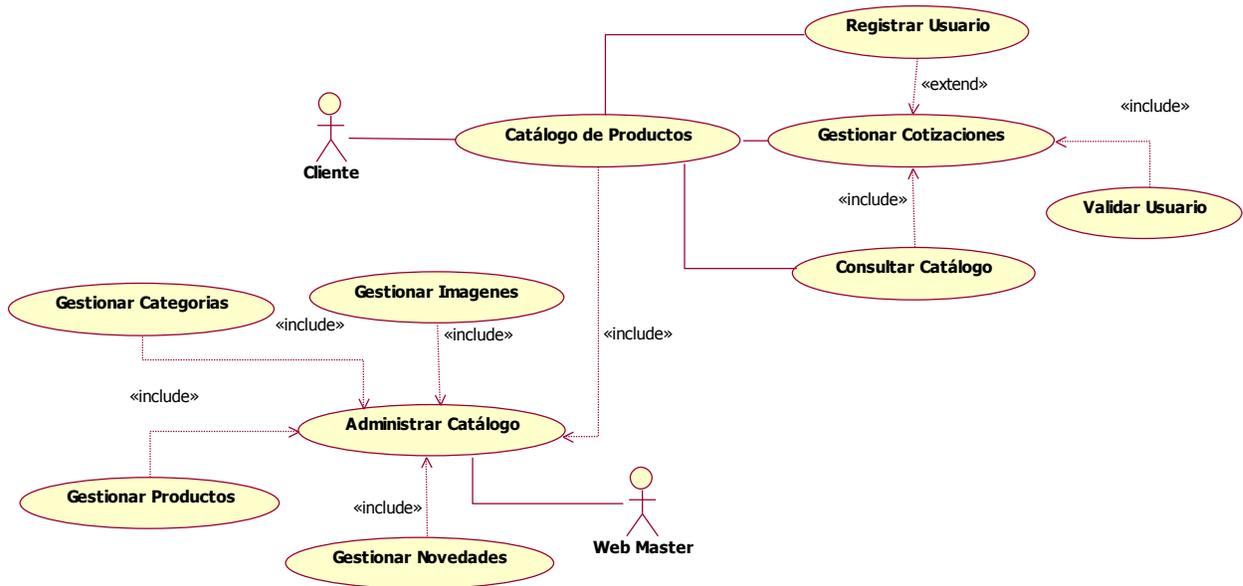
3.6 Diagramas de Casos de Uso

DIAGRAMAS CASOS DE USO

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight(WPF/E) y XAML”

Versión 1.2.0

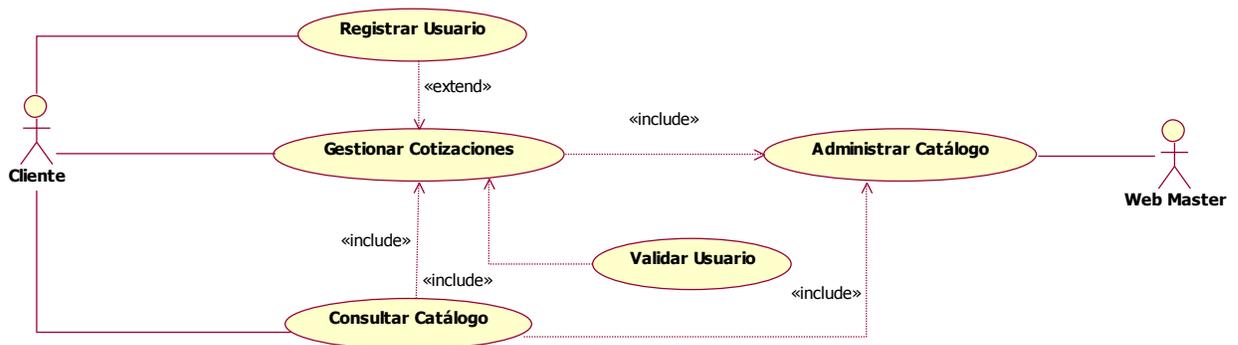
3.6.1 Diagrama de casos de uso General



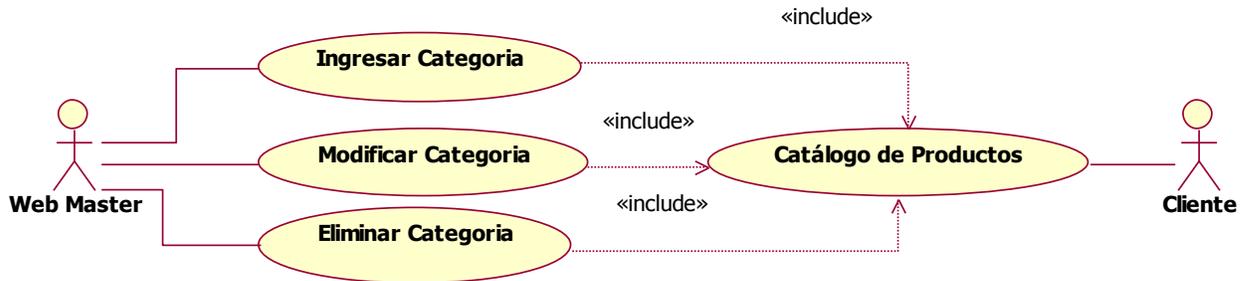
Para información detallada de cada caso de uso, se recomienda ver los documentos de “Especificaciones de Casos” de uso detallados en Anexos.

3.6.2 Diagrama de casos de uso por módulos

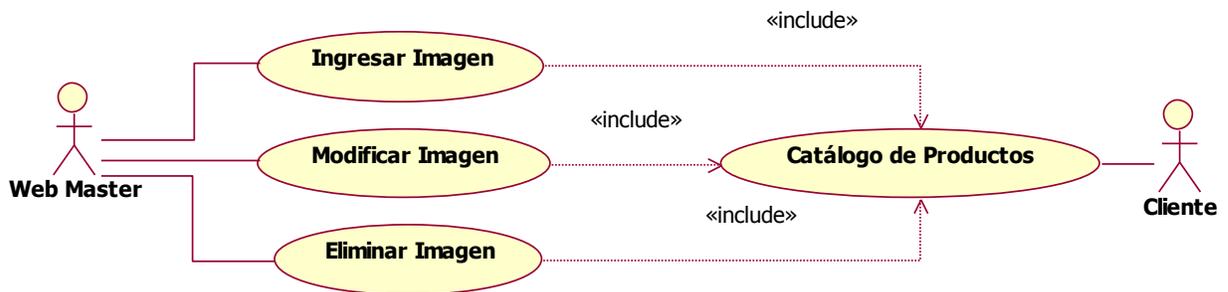
Módulo Administrar Catálogo



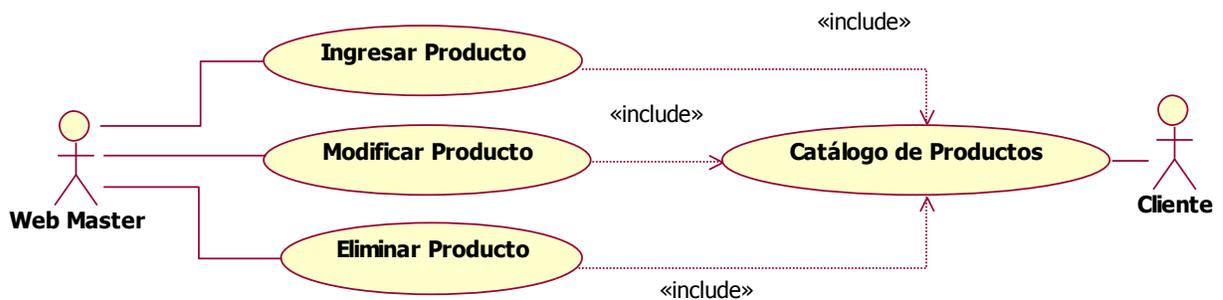
Sub módulo Gestionar Categorías



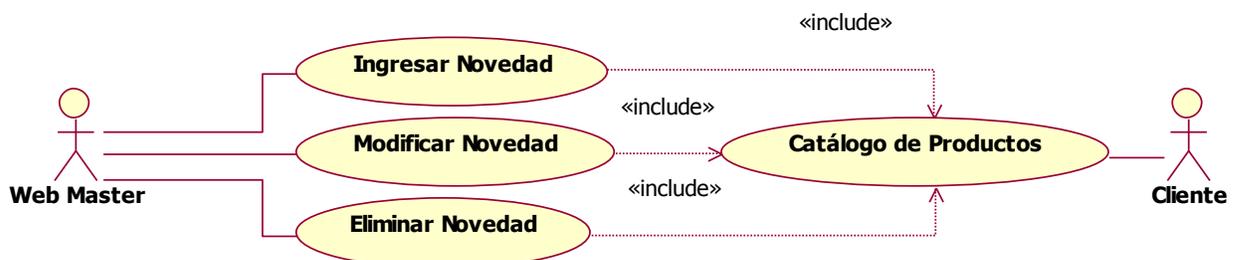
Sub módulo Gestionar Imágenes



Sub módulo Gestionar Producto



Sub módulo Gestionar Novedades



3.7 Diagramas de Secuencia Principales

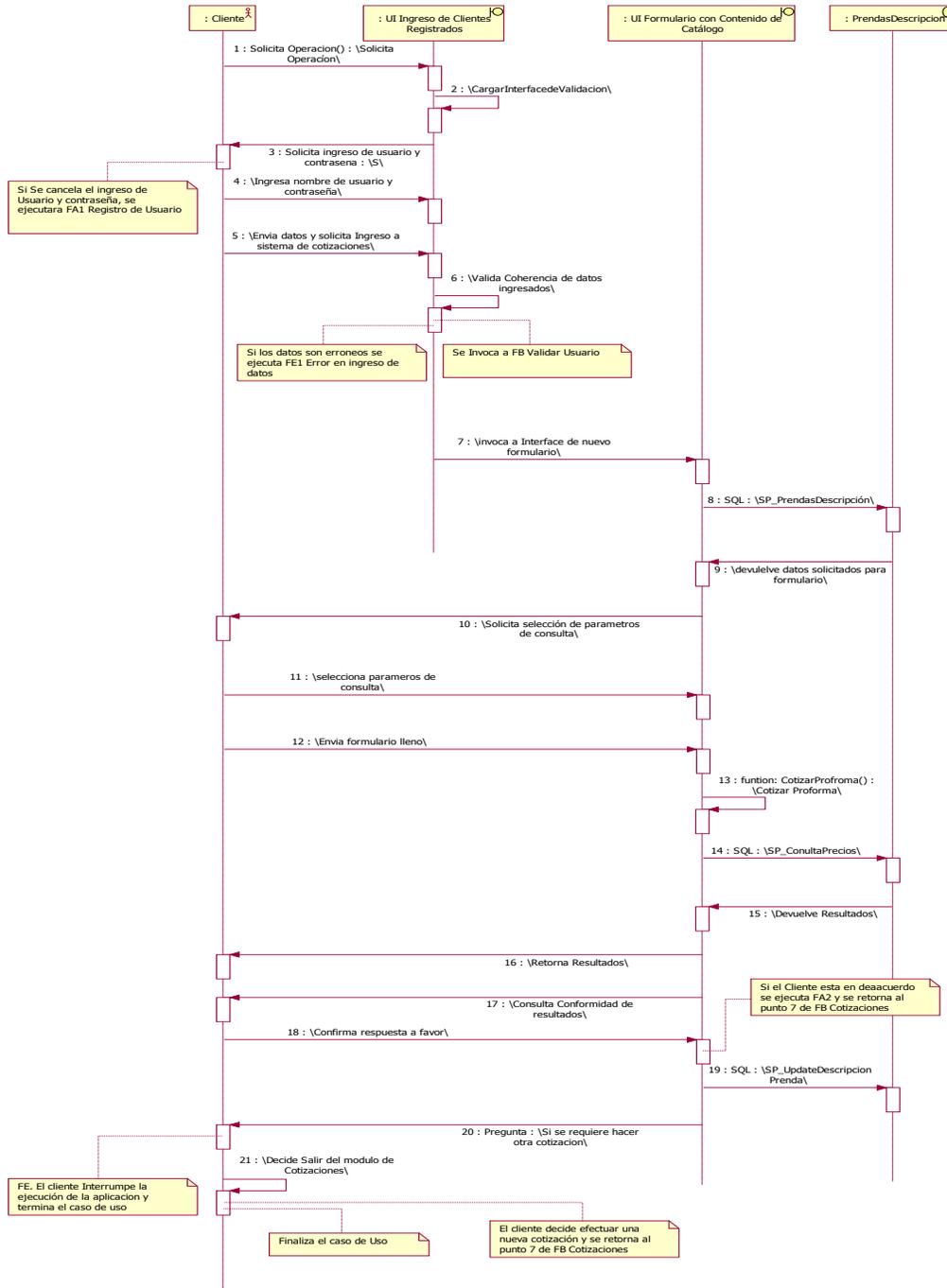
DIAGRAMAS DE SECUENCIA PRINCIPALES

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

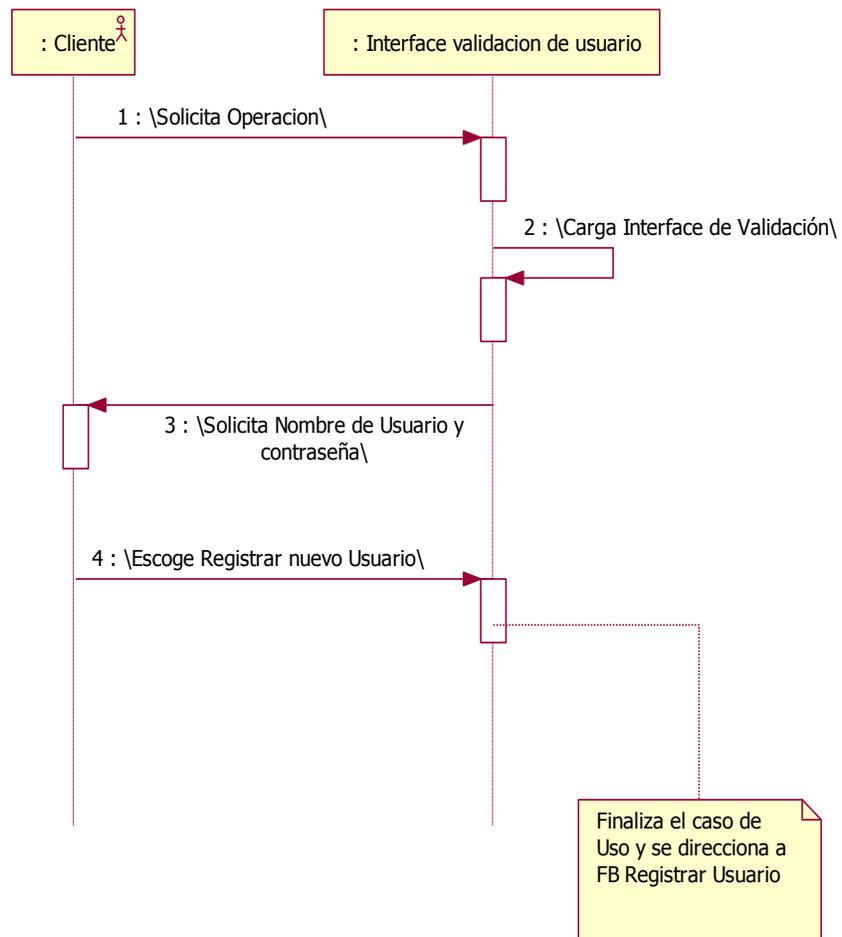
Versión 1.2.0

3.7.1 Gestionar Cotizaciones

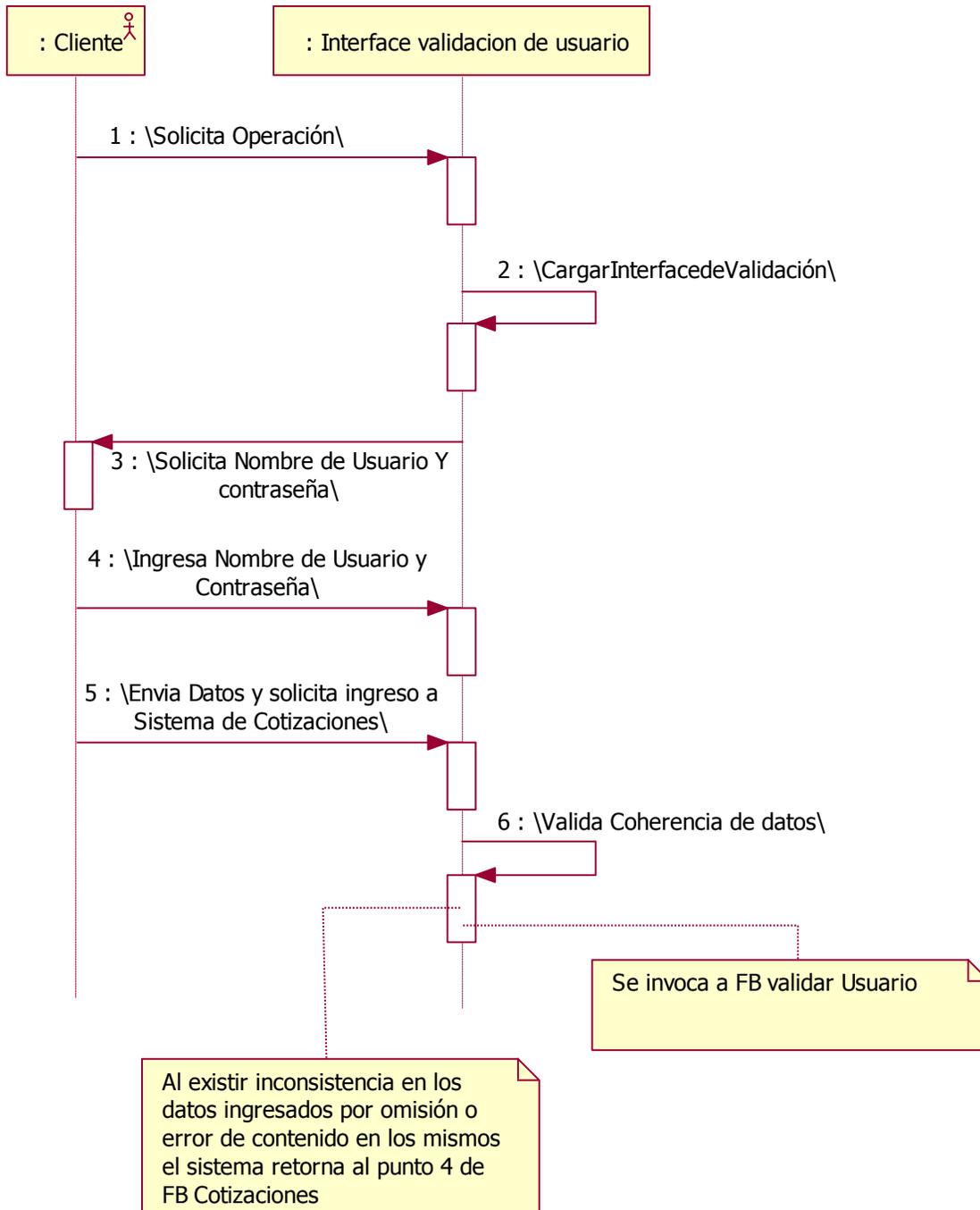
Flujo Básico: Gestionar Cotizaciones



Flujo Alternativo: FA1 Se desea registrar un Nuevo Usuario



Flujo de Excepción:



3.8 Diagramas de Secuencia Secundarios

Los diagramas de secuencia secundarios se detallan en el anexo “C”

3.9 Diagramas de Colaboración de Análisis

Los diagramas de colaboración de análisis se detallan en el anexo “D”.

3.10 Diagramas de Colaboración de Diseño

Los diagramas de colaboración de diseño se detallan en el anexo “E”.

3.11 Diagramas de Estado

Los diagramas de estado secundarios se detallan en el anexo “F”.

3.12 Diagramas De Clases

DIAGRAMAS DE CLASES

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

Versión 1.2.0

Diagrama De Clases De Análisis

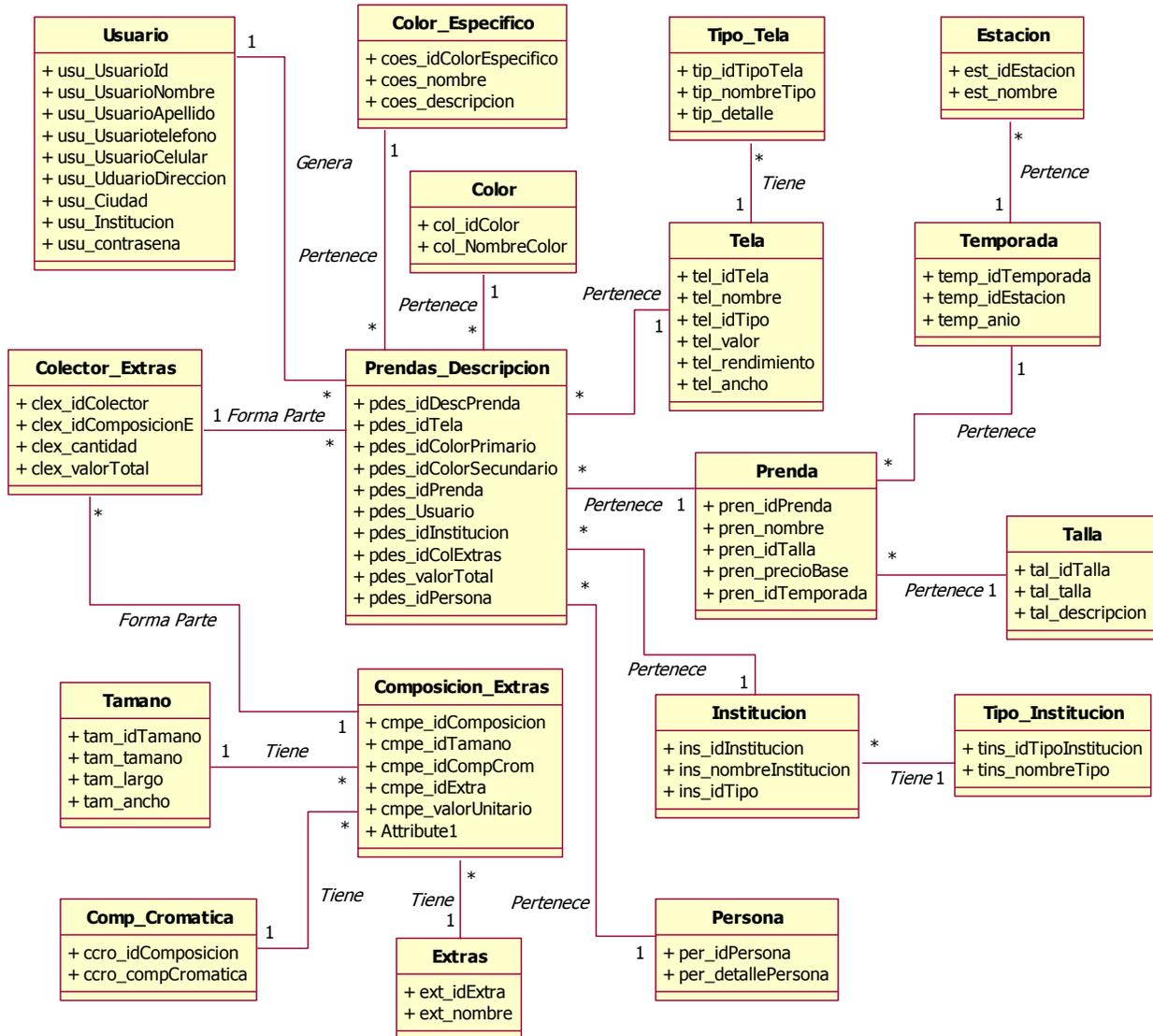
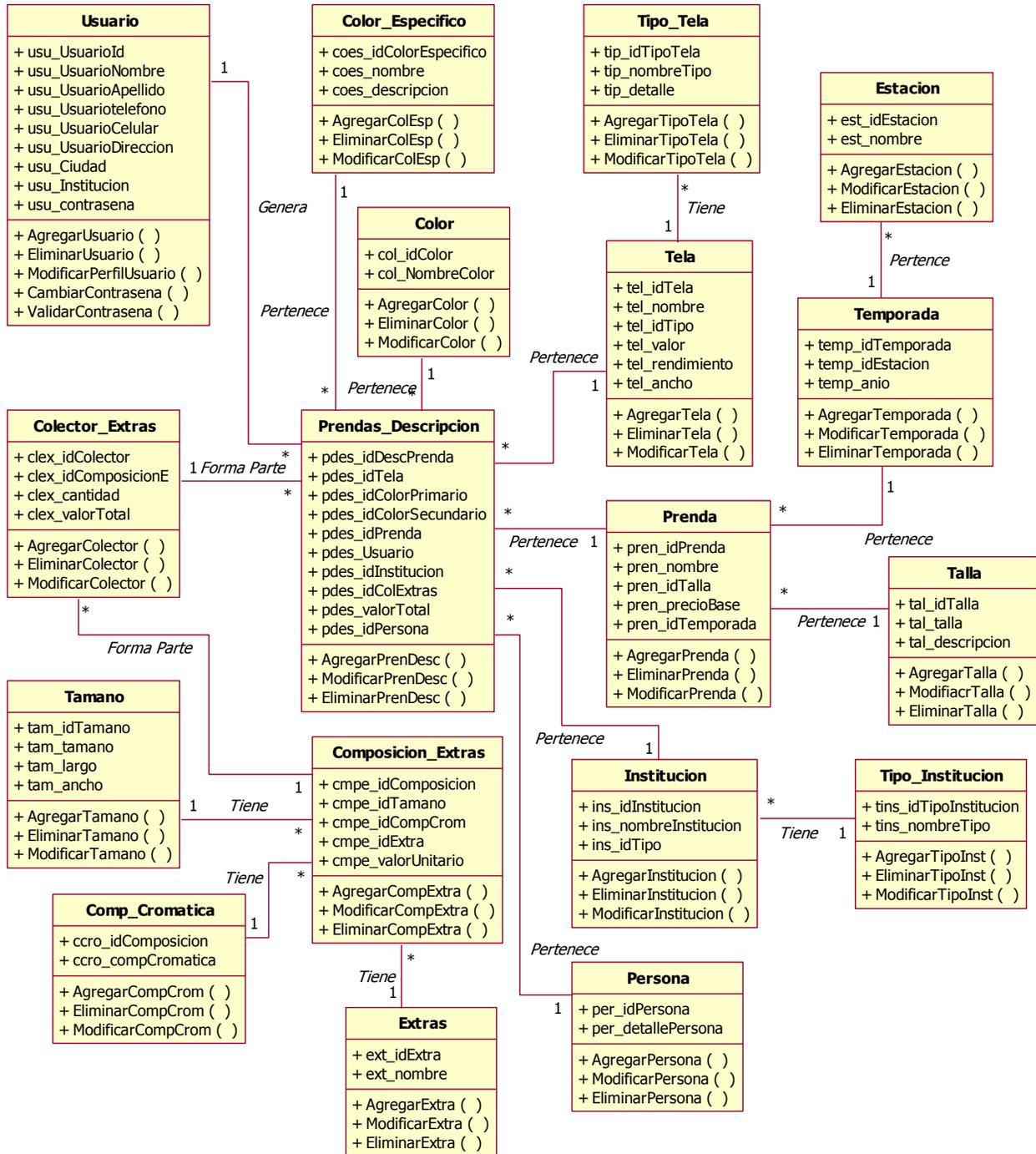


Diagrama De Clases De Diseño



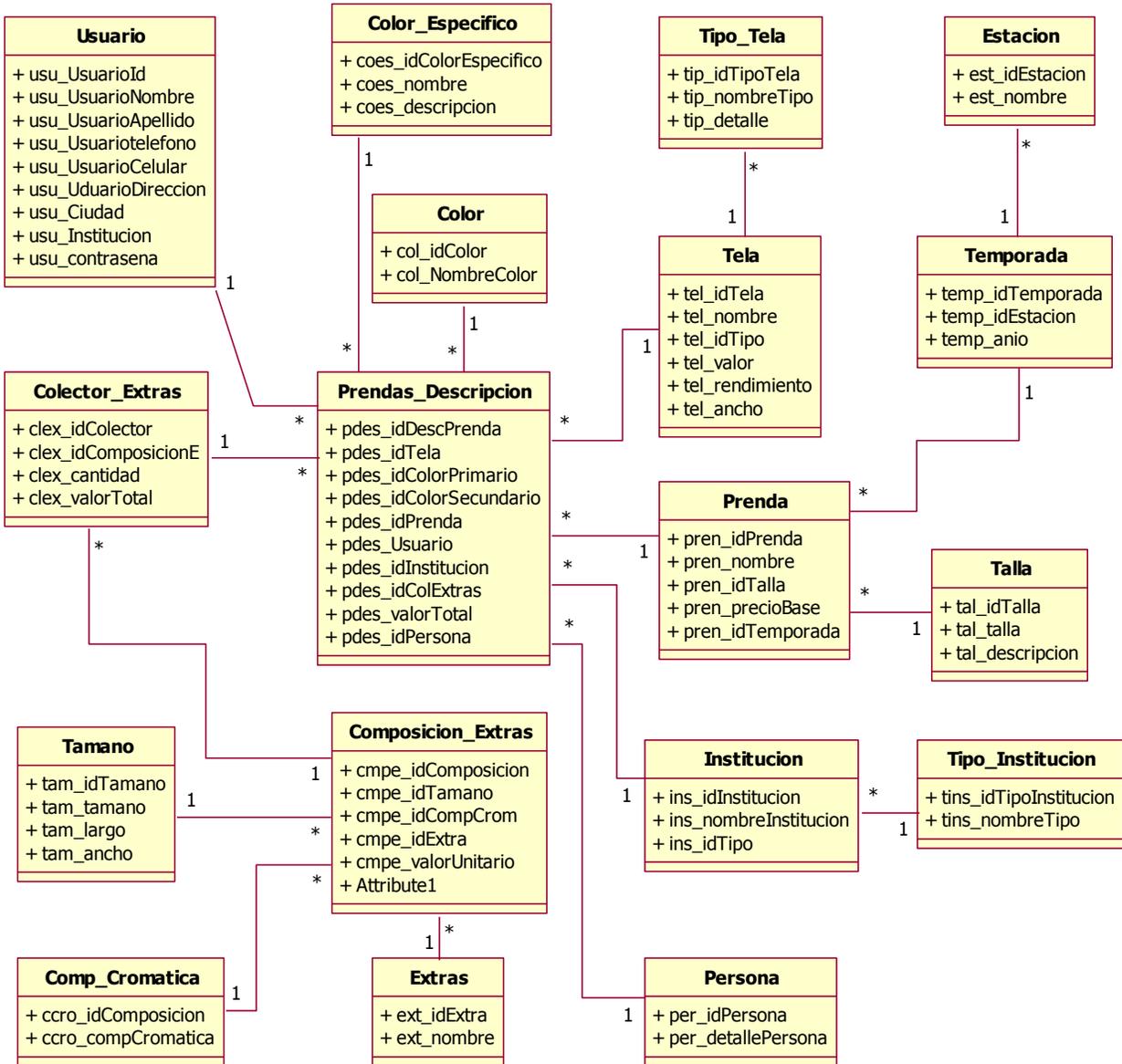
3.13 Modelo De Datos

MODELO LOGICO

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

Versión 1.2.0

Modelo Lógico



MODELO FISICO

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight(WPF/E) y XAML”

Versión 1.2.0

3.14 Implementación de código

La Implementación de código se detalla en el anexo “G”

3.15 Diagrama de componente

El diagrama de componente se detalla en el anexo “H”

3.16 Diagrama de despliegue

El diagrama de despliegue se detalla en el anexo “I”

3.17 Resumen De Casos De Pruebas

Se realizaron los casos de pruebas del sistema los cuales se detallan en el anexo “J”.

3.18 Glosario De Términos

El glosario de términos del sistema se encuentra detallado en el anexo “K”

3.19 Análisis de costos del proyecto

Todo catálogo requiere una continua actualización de contenido, más aún cuando el nicho de mercado al cual atiende tiene que ver con el área textil específicamente con prendas de vestir como lo son los uniformes.

Los continuos cambios de estilo de prenda determinados por tendencias de moda, temporadas, introducción de nuevos modelos de prendas y materias primas e incluso factores variantes como son los precios y costos de producción, obligan a mantener actualizada la variedad de uniformes que se exponen, manteniendo a su vez, una imagen actual, atractiva y real la que satisfaga las expectativas de productores y consumidores.

Todos estos factores tiene un costo que afecta directamente al sistema; por tanto, el método empleado para la estimación de costos será el método Mark II, según este método se deberá realizar las siguientes actividades:

- Identificación de todas las funciones.
- Identificación de todas las entidades, tipos de datos de entrada y tipos de datos de salida.
- Cálculo de los puntos función no ajustados.
- Valoración de grados de influencia.
- Ajuste de complejidad técnica.
- Obtención del tamaño de las partes en línea y por lotes.
- Cálculo del tamaño total del sistema.

- Cálculo de la productividad estimada.
- Cálculo del esfuerzo en horas.
- Cálculo de la tasa del tiempo de entrega del proyecto.
- Cálculo de plazo de entrega.
- Descomposición en fases.

Cálculo de los puntos de función no ajustados

La siguiente tabla permitirá la valoración de los puntos de función no ajustados de las funciones que intervienen en el sistema

	Catálogo	Cotizaciones
# Entidades	18	11
# Entradas	37	22
# Salidas	19	11

Siguiendo el método el número de entidades de las funciones se multiplica por 1,66, el número de entradas por 0,58 mientras que el número de salidas por 0,26, una vez realizadas estas operaciones obtendremos la siguiente tabla:

	F1	F2
# Entidades	29.88	18.26
# Entradas	21.46	12.76
# Salidas	4.94	2.86
NPF no ajustados	56.28	33.88

Puntos ajustados de las funciones por lotes:

PFNA b = 0

Puntos no ajustados de las funciones en línea

PFNA_o = 90.16

Valoración de grados de influencia

El objetivo de esta etapa es realizar la adaptación de la estimación a las condiciones de trabajo mediante las cuales se ha de desarrollar el sistema.

Mark II tiene 19 atributos de ajuste, a cada uno de ellos se le asigna un valor de 0 a 5, dependiendo del grado de influencia de estos. A continuación se presenta los valores asignados a los atributos de ajuste

ATRIBUTOS		INFLUENCIA
1	Comunicación de datos	3
2	Funciones distribuidas	3
3	Prestaciones	2
4	Gran uso de la configuración	4
5	Velocidad de las transacciones	5
6	Entrada de datos en línea	4
7	Diseño para la eficiencia del usuario final	5
8	Actualización de datos en línea	3
9	Complejidad del proceso lógico interno de la aplicación	5
10	Reusabilidad del código	5
11	Facilidad de instalación	3
12	Facilidad de operación	4
13	Localizaciones múltiples	1
14	Facilidad de cambios	4
15	Requerimientos de otras aplicaciones	1
16	Seguridad, privacidad, auditabilidad	2
17	Necesidades de formación	1
18	Uso por terceras partes	2
19	Documentación	3
Suma TGI		60

Ajuste por complejidad técnica

Una vez obtenido el valor de los atributos, se realiza la suma de estos y se obtiene un valor entre 0 y 95, a partir de cual se obtendrá el factor de ajuste.

$$ACT = 0,65 + 0,005 * TGI$$

Siendo:

ACT: Ajuste por Complejidad Técnica.

TGI: Total Grados de Influencia.

Remplazando los valores obtenidos en la fórmula obtenemos:

$$ACT = 0,65 + 0,005 * TGI$$

$$ACT = 0,95$$

Obtención del tamaño de las partes en línea y por lotes

A continuación hay que ajustar los puntos función para cada una de las partes, por lotes y en línea, mediante la aplicación de las siguientes fórmulas:

$$PFAb = PFNAb * ACT$$

$$PFAo = PFNAo * ACT$$

Siendo:

PFAb: Puntos Función ajustados de las funciones por lotes

PFNAo: Puntos Función no ajustados de las funciones por lotes

PFAo: Puntos Función ajustados de las funciones En línea

PFNAo: Puntos Función no ajustados de las funciones En línea

ACT: Ajuste por Complejidad Técnica.

Los valores obtenidos en el sistema son:

PFA b = 0

PFA o = 85,65

Cálculo del tamaño total del Sistema

Seguidamente, se calculará el tamaño total, en Puntos Función, del sistema para lo que se aplicará la siguiente fórmula:

$$PFA = PFAb + PFAo$$

Donde:

PFA: Tamaño del Sistema completo en Puntos Función

PFAb: Tamaño de la parte Por lotes en Puntos Función

PFAo: Tamaño de la parte En línea en Puntos Función

El valor de esta etapa para el sistema será:

$$PFA = 85.65$$

Cálculo de la productividad estimada

Para el cálculo de la productividad estimada, es necesario aplicar la siguiente fórmula:

$$P = A \left[0,11e^{-[(S-250)/575]^2} + \frac{0,01S^{1,1}}{522} \right]$$

Siendo:

P: Productividad

A: Media de la Industria informática:

A= 1,0 para 3GL

A= 1,6 para 4GL

S: Tamaño del Sistema en PFA

El valor de esta etapa para el sistema es:

P = 0,12

Cálculo del esfuerzo en horas

Una vez conocida la productividad estimada, habría que calcular el esfuerzo en horas de trabajo, para lo que se aplica la siguiente fórmula:

$$W = \frac{B \cdot PFA}{P}$$

Siendo:

W: Esfuerzo en horas de trabajo

B: Factor de complejidad

B= 1,0 si es en línea

B= 1,5 si es por lotes

$B = (S_o + 1,5 S_b) / (S_o + S_b)$, si el sistema es mixto

PFA: Puntos Función ajustados

P: Productividad en PF/hora

El valor obtenido es:

$$W = 702.50$$

Cálculo del plazo de entrega

En primer lugar habría que calcular el factor a aplicar, estando éste en relación directa con el tamaño del sistema y cuyo valor se obtiene mediante la aplicación de la siguiente fórmula:

$$E = 0,45 \sqrt{S}$$

Siendo:

E: Puntos Función / semana

S: Tamaño del Sistema en PFA

El valor obtenido por la fórmula es:

$$E = 4.16$$

$$PE \text{ (semanas)} = 20.57$$

Porcentaje de las etapas del proyecto.

Análisis	35%
Diseño	24%
Codificación	30%
Pruebas	10%
Capacitación	1%

Periodo de duración de cada etapa en semanas

$$\text{Análisis} = 7.20$$

$$\text{Diseño} = 4.94$$

$$\text{Codificación} = 6.17$$

$$\text{Pruebas} = 2.06$$

$$\text{Capacitación} = 0.21$$

Costo de la etapa de análisis por persona.

Cargo	Honorario mensual (USD)	Honorario semanal (USD)	Persona	Análisis		Diseño		Codificación		Pruebas		Capacitación		Subtotales (USD)
				%	USD	%	USD	%	USD	%	USD	%	USD	
ANALISTA	500.00	125.00	Juan David Utreras		0.00	100	616.99		0.00		0.00		0.00	616.99
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
PROGRAMADOR	500.00	125.00	Juan David Utreras		0.00		0.00	100	771.24		0.00		0.00	771.24
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
PROBADOR	300.00	75.00	Juan David Utreras		0.00		0.00		0.00	100	154.25		0.00	154.25
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
DOCUMENTADOR	500.00	125.00	Juan David Utreras		0.00		0.00		0.00	50	128.54	10	2.57	131.11
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
CONTROL DE CALIDAD	100.00	25.00	Juan David Utreras		0.00		0.00		0.00	70	35.99		0.00	35.99
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
		0.00			0.00		0.00		0.00		0.00		0.00	0.00
Subtotales (USD):				0.00		616.99		771.24		318.78		2.57		
TOTAL (USD):								1,709.57						
Porcentajes (%)					0.00		36.09		45.11		18.65		0.15	

Total Costo de esfuerzo 1709.57

Costo de hardware y software del proyecto.

Hardware	750
Software	1600
Consumibles	50

Costo total del proyecto.

Total Costo de esfuerzo = 1709.57

Costo de Hardware y software del proyecto = 2350

Costo total sin utilidad= 4059.57

Se tendrá un 10% de ganancia en la construcción de este software por lo tanto el costo total del software será:

Costo total con ganancia = 4465.53

CAPÍTULO IV

CONCLUSIONES

Concluido el desarrollo del proyecto de “Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”, se establecen las siguientes conclusiones:

- La necesidad del sistema de catálogo en línea para la empresa Donaire Needle Point, se da como resultado a un proceso de globalización que exige a las empresas en general, el uso de herramientas tecnológicas actuales.
- El sistema de cotización mediante catálogo en línea es un recurso versátil que beneficia a los clientes reales, potenciales; y de manera específica a la empresa Donaire Needle Point.
- Entre los beneficios verificados se establecen los siguientes:
 - Refuerzo de Imagen corporativa.
 - Promoción efectiva de productos.
 - Optimización en tiempo de consultas.
 - Registro efectivo de Clientes.
 - Consulta inmediata de cotizaciones.
 - Mejor aprovechamiento de los recursos de la empresa.

Por ende se anticipan mayores beneficios para la empresa

- La tecnología Silverlight; implementada en este proyecto; demuestra ser útil y compatible con HTML, AJAX, Web Forms de ASP.NET y puede coexistir sin problema en entornos que incluyan contenido Flash.
- Silverlight es una tecnología prometedora y robusta que presenta nuevos estándares de generación de software rico en contenido visual orientado a la Web; y como tal tiene grandes virtudes como la reutilización de código; el uso común con WPF de código XAML; entre otros. Pero lamentablemente es una tecnología que aún se encuentra en proceso de desarrollo por lo que al momento posee limitantes que impiden un desarrollo versátil y vistoso; a diferencia de Adobe Flash que al ser el competidor con más trayectoria y uso; supera en gran manera muchas funcionalidades de esta nueva tecnología de Microsoft.
- La metodología RUP escogida para el desarrollo de este sistema permite adaptarse a cambios así como a nuevos requisitos o requerimientos de una manera sencilla y ágil. Adicionalmente proporciona una amplia comprensión a los usuarios acerca de cómo se realiza el sistema en sus fases de construcción e implementación.
- El código empleado en cada una de las tecnologías no se mezcla, por lo que no se pierde legibilidad de interpretación y trabajo conjunto entre tecnologías.

CAPÍTULO V

RECOMENDACIONES

- Se recomienda el empleo de la tecnología de Silverlight a partir de la versión 2.0 debido a que la versión 1.0; contiene aun muchas limitantes.
- Silverlight como tal tiene notables ventajas frente a Adobe Flash en lo referente a “video encoding” y a “video release”; ya que el motor simulador de Preloader de Silverlight es mucho más liviano y efectivo al momento de ejecutar archivos de extensión AVI, MPG e incluso MP3 y MP4; por la versatilidad ofrecida mediante XAML.
- Se recomienda el empleo de Silverlight para trabajar directamente con ASP.NET ya que los “Web Forms” de esta tecnología pueden ser potenciados mediante el elemento el “Canvas” del “Add on” de Silverlight; el mismo que despliega su contenido a modo de capa base. Permitiendo de esta forma colocar componentes de ASP.NET sobre el contenido de Silverlight a modo de “escritorio” sin interferir en el despliegue de los mismos, ni en su funcionalidad aplicativa.
- Para determinados casos, se recomienda usar de la tecnología Adobe Flash; esto es, en lo referente a elaboración de interfaces de usuario y presentaciones más complicadas y con contenidos interactivos complejos; especialmente en lo referente a multimedia. Esto se debe a que Adobe Flash incluye un motor generador de proyectores de publicación de tipo

EXE. Atributo que ni Windows Presentation Foundation ni Silverlight han desarrollado aun hasta las presentes versiones.

CAPÍTULO VI

BIBLIOGRAFIA

6.1 Recursos Web

- <http://www.silverlight.net>
- <http://www.erain.com>
- <http://www.templatemonster.com>
- <http://www.asp.net>
- <http://www.asp.net/ajax>
- <http://www.microsoft.com/expression>
- <http://www.microsoft.com/spanish/msdn/articulos/archivo/010507/voices/bb404300.msp>
- <http://msdn2.microsoft.com>
- <http://es.wikipedia.org/wiki>
- <http://www.desarrolloweb.com/asp>
- <http://www.informatizate.net>
- <http://www.rational.com>

6.2 Recursos Textuales

- Grady Booch ; James Rumbaugh; Ivar Jacobson, “The Unified Modeling Language User Guide”, Addison Wesley, Primera Edición, 1998
- Grady Booch ; James Rumbaugh; Ivar Jacobson, “The Unified Modeling Language Reference Manual”, Addison Wesley, Segunda Edición, 2004
- Grady Booch; “Object-Oriented Analysis and Design with Applications”, Segunda Edición
- Jim Conallen Building; “Web Applications with UML” Second Edition, Addison Wesley 2002
- Nathan, Adam ; “Silverlight 1.0 Unleashed” ; Sams 2007.
- Nathan, Adam ; “Windows Presentation Foundation Unleashed” ; Sams 2007.
- Laurence Moroney; “Foundations of WPF, An Introduction to Windows Presentation Foundation” ; Apress 2006

CAPÍTULO VII

ANEXOS

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

Versión 1.2.0

7.1. Anexo “A”: Especificaciones Suplementarias

ESPECIFICACIONES SUPLEMENTARIAS

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight(WPF/E) y XAML”

Versión 1.2.0

7.1.1 Propósito

El propósito de detallar las Especificaciones Suplementarias; es describir todas las funcionalidades del sistema, que no son necesarias describir como casos de uso; para poder explicar de una manera más detallada la funcionalidad del sistema para que sea fácil de comprender para el usuario.

7.1.2 Alcance

En este documento se detallan las funciones del sistema que no están incluidas como casos de uso, con lo cual se detalla las características restantes del sistema.

7.1.3 Vista General

Se describe requerimientos de funcionalidad, usabilidad, confiabilidad, desempeño, y soporte. Luego, se describe ciertas restricciones de diseño, interfaces que usará el sistema, y algunas reglas de negocio.

7.1.4 Funcionalidad

7.1.4.1 Consultar Catálogo

Los clientes al ingresar al catálogo de la empresa Donaire Needle Point, podrán consultar:

- Prendas predefinidas; clasificadas según la categoría a la que pertenecen para hombres, mujeres, niños y niñas.
- Uniformes de instituciones clientes de la empresa.
- Cotizaciones generadas por el cliente según los parámetros escogidos.
- Novedades referentes a nuevas líneas de productos que la empresa lanzará a futuro.

- Productos complementarios como bufandas, gorras y ponchos.
- Información relativa a tendencias de moda según las temporadas
- Información complementaria como tonalidades de color.

7.1.4.2 Cargar el contenido según la categoría.

El sistema deberá cargar el contenido referente a las categorías seleccionadas por el usuario.

Por ejemplo; si el usuario escoge la categoría de uniformes, será necesario que el sistema direcciona la petición del usuario al subsistema que carga la información relativa a uniformes.

7.1.4.3 Registrar Clientes.

Los usuarios del sistema para hacer uso del subsistema de cotización de uniformes deberán registrar sus datos de manera obligatoria por una sola vez. Para lo cual el sistema direccionará al cliente a una página de registro que almacenará los datos del cliente de forma permanente.

7.1.4.4 Crear contraseñas de usuario para el acceso

Los usuarios deberán estar registrados para poder hacer uso del sistema de cotización de uniformes personalizados.

Para esto se definirá como contraseña la cédula de ciudadanía del usuario registrado. Una vez que el usuario registre sus datos; la contraseña estará registrada y podrá ser empleada en el sistema en cualquier momento.

Para esto la contraseña deberá ser de digitada mediante dígitos numéricos.

7.1.5 Usabilidad

El Sistema debe ser amigable

En vista de que el sistema esta direccionado a un público en general; se requiere que el mismo sea fácil de usar, y que en lo posible reduzca el esfuerzo de adaptación tecnológica por parte de los clientes que en muchos casos no disponen de conocimientos previos de este tipo de aplicaciones.

Para ello el sistema debe permitir que el cliente sea direccionado a su objetivo de búsqueda sin mayor complicación. Dando lugar a una navegación entre páginas de una forma ágil y efectiva mediante menús.

Guías sobre campos complejos

Con el fin de que se tenga una óptima comprensión del sistema a construir, se realizará un documento que desglose todos los términos usados en el desarrollo del sistema que necesiten explicación el que se encuentra en el glosario de términos.

El mismo proporcionará al usuario una explicación corta y clara de cada término involucrado en el desarrollo del sistema con el objetivo de que las personas involucradas obtengan una percepción clara de la formulación del mismo.

Apariencia de las interfaces

La interface de usuario debe ser amigable y debe detallar de forma funcional todas las características relativas al producto permitiendo que el cliente encuentre de manera atractiva toda la información que busca.

La apariencia de las interfaces es muy importante debido a que el usuario va a interactuar con ellas. Por tanto el sistema requiere interfaces que sigan normas que permitan un fácil manejo por parte del usuario.

Las interfaces desarrolladas permiten presentar y manipular los objetos e información de la plantilla, son interfaces intuitivas lo que significa que las interfaces serán fáciles de aprender y comprensibles para el usuario.

Las interfaces son estéticamente agradables y tienen claridad visual, conceptual y lingüística.

Mensajes de error

Cada mensaje es desplegado al usuario de manera clara y sutil; de manera que el usuario pueda entender de qué se trata el error evitando que entre en pánico ante el evento.

El mensaje de error notifica si el sistema detecta un error, a la vez de notificar otro tipo de errores que podrían suceder como resultado de acciones iniciadas por parte del usuario.

Los mensajes del sistema utilizan el formato de “cuadro de mensaje modal” el cual utiliza un pequeño cuadro de diálogo que aparecerá sobre la ventana principal o sobre el cuadro de diálogo en los que esté interactuando el usuario.

Además, este formato de mensaje requiere que el usuario realice una acción para poder continuar usando la aplicación.

Acceso 365/24

El sistema tiene que estar disponible cuando se lo necesite para satisfacer las necesidades de los clientes de Donaire Needle Point (DNP); para lo cual el sistema debe permitir el acceso a los usuarios en todo momento 365 días al año las 24 horas al día.

7.1.6 Desempeño

Mantener los niveles de velocidad

El tiempo de respuesta entre el usuario y la aplicación; es un factor importante. Es necesario considerar agilizar operaciones de modo que esta cantidad de tiempo de respuesta sea razonable; tomando en cuenta que el usuario valora de modo distinto el tiempo que el sistema tarda en responder a una acción.

7.1.7 Soporte

Ambientes de prueba

El sistema cuenta con ambientes iguales al ambiente de producción, para realizar pruebas, siguiendo documentos guías para poder realizar dichas pruebas.

Soporte de Usuarios

Se recomienda que los usuarios se pongan en contacto con el Webmaster de la empresa Donaire Needle Point; si existe algún problema de interacción con el sistema.

7.1.8 Restricciones de diseño

Lenguajes de programación

Se debe tomar en cuenta que se deben utilizar los lenguajes Action Script 2.0, XML, XAML, Java Script, ASP.NET, AJAX, HTML; para la codificación del sistema.

Base de datos

Se debe tomar en cuenta que las bases de datos para uso del sistema está desarrollada en Microsoft SQL2005.

7.1.9 Interfaces

Interfaces de hardware

Ninguna.

Interfaces de software

No existen interfaces de software para este sistema.

Interfaces de comunicación

No existen interfaces de comunicación para este sistema.

7.2. Anexo “B”: Especificaciones de casos de uso

MODELO Y ESPECIFICACIONES DE CASOS DE USO

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

Versión 1.2.0

7.2.1 Registrar Usuario

Breve Descripción

Este caso de uso muestra la funcionalidad del sistema cuando un usuario desea registrar sus datos; para poder acceder a las utilidades del sistema como lo es el “Módulo de Cotizaciones”

Precondiciones

Ingreso de datos válidos

Los datos ingresados por el cliente deben ser correctos y completos.

Pos condiciones de éxito

Ingreso de datos al sistema

El usuario registra sus datos y recibe una contraseña de acceso al sistema.

Pos condiciones de falla

Los datos del usuario contienen errores.

El usuario digito sus datos omitiendo caracteres o campos; o a su vez sobrepaso el límite permitido en cada campo.

Cliente no puede cargar sus datos debido a una falla en la red

El usuario no puede ingresar al sistema a causa de una falla en la red.

Actor Principal

Usuario.

Flujo de Eventos

Flujo Básico

Si el interesado solicitó registrar un nuevo usuario, se ejecutarán las siguientes operaciones:

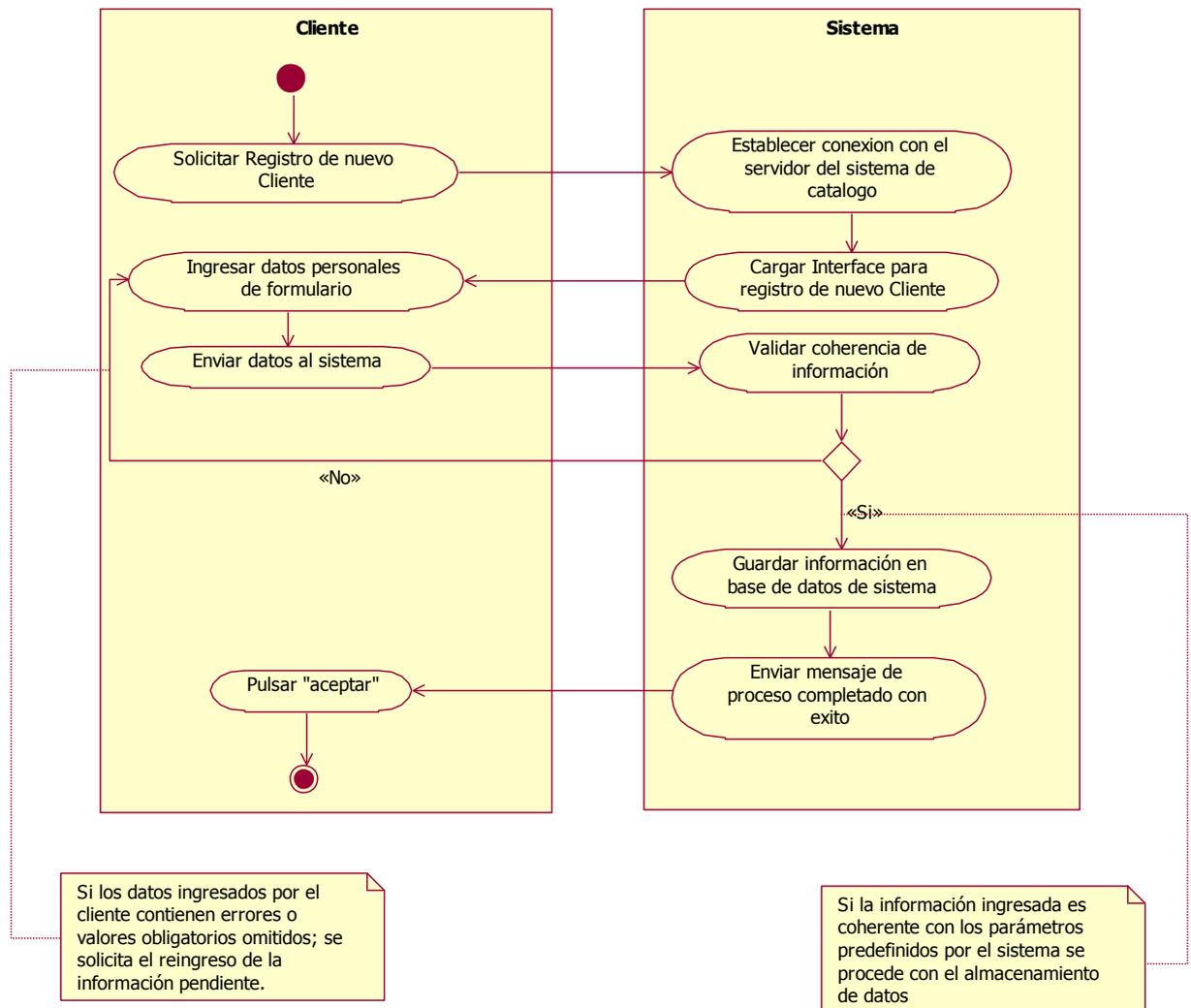
1. El sistema establece conexión con el servidor del sistema de Catálogo.
2. El sistema carga la interface para registro de Cliente y solicita el ingreso de datos personales de nuevo usuario.
3. El usuario ingresa datos personales.
 - a. FE1 El usuario solicita cancelar el ingreso.
 - b. Finaliza caso de uso.
4. El usuario solicita enviar datos al sistema.
 - a. FE2 El usuario no envía correctamente sus datos personales
 - b. Finaliza caso de uso.
5. El sistema efectúa validación de coherencia en la información ingresada.
 - a. FA El usuario no ingresó la información completa o correcta para la creación de "nuevo usuario".
 - b. El sistema notifica que se debe ingresar la información coherente para la creación de nuevo usuario.
 - c. Regresa al paso 3 del sub flujo.
6. El sistema guarda información en la base de datos del sistema.
7. El sistema envía mensaje de "registro completado exitosamente".
8. El usuario acepta el resultado emitido por el sistema.
9. Finaliza el caso de uso.

Requerimientos Especiales

No existen requerimientos especiales.

Diagrama

Outline Registrar Usuario



Interfaz gráfica con el usuario

Pantalla 1 de Registro de Usuario



Pantalla 2 de Registro de Usuario



Relaciones

Interfaces con otros sistemas

No existen interfaces con otros sistemas.

Inclusiones

No existen.

Extensiones

No existen extensiones.

7.2.2 Consultar Catálogo

Breve Descripción

Este caso de uso muestra la funcionalidad del sistema cuando un usuario desea efectuar consultas al catálogo por categorías predefinidas.

Precondiciones

Conectividad con la aplicación Catálogo

Tanto el servicio de red como la conexión con la base de datos deben estar activos.

Disponibilidad de Catálogo

El contenido por categorías de catálogo debe estar disponible.

Pos condiciones de éxito

Consulta Efectuada

El usuario puede acceder a la información de catálogo por categorías y puede consultar uno o más productos requeridos.

Pos condiciones de falla

No se carga información de catálogo total o parcialmente

El contenido de catálogo no está disponible, total o parcialmente.

Falla en la red

El usuario no puede ingresar al sistema a causa de una falla en la red.

Actor Principal

Usuario.

Flujo de Eventos

Flujo Básico

Si el usuario solicitó ingresar a consultas a catálogo de productos, se ejecutarán las siguientes operaciones:

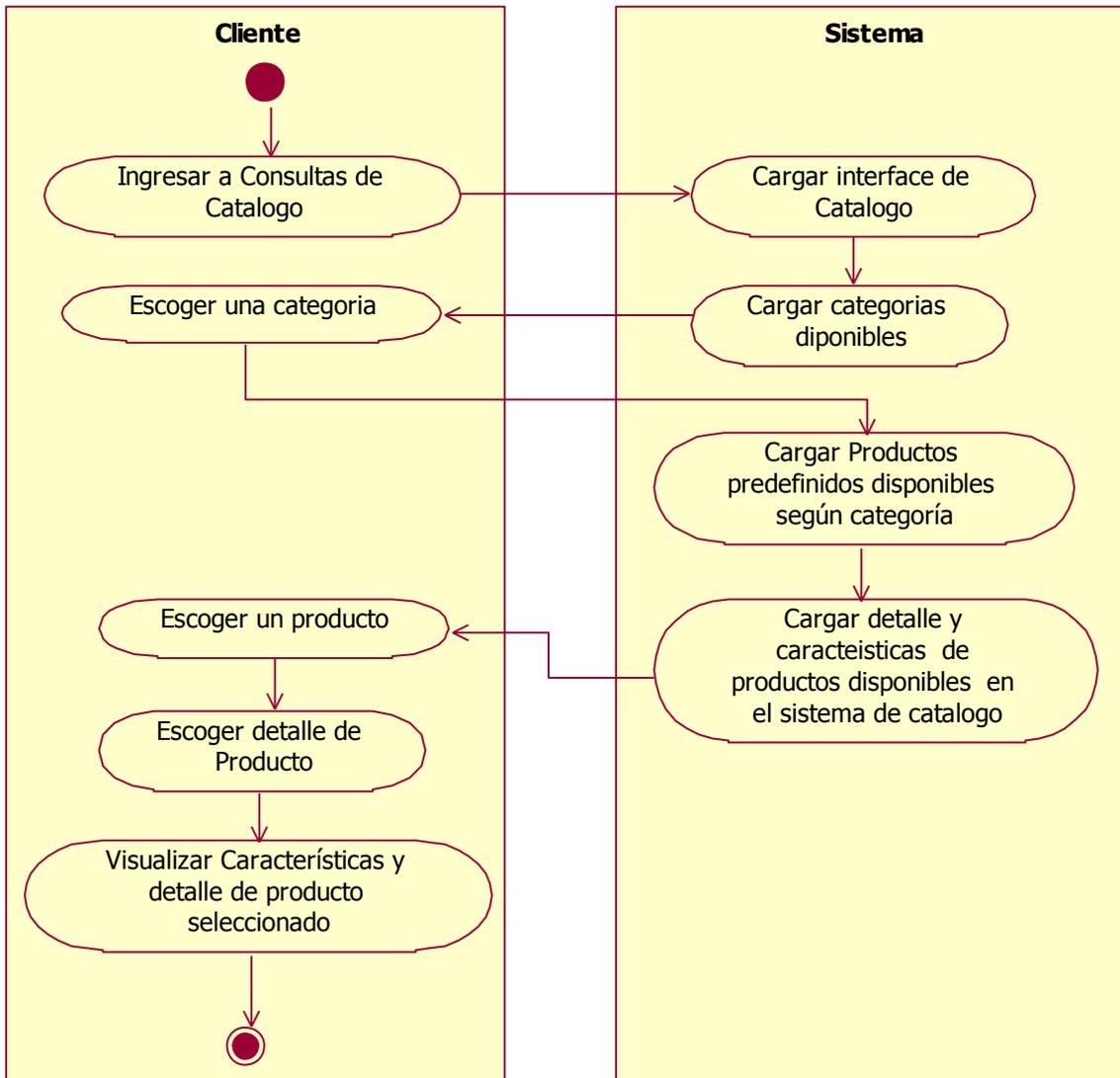
1. El sistema carga interface de catálogo de productos
2. El sistema carga categorías disponibles
3. El usuario selecciona una de las categorías disponibles
 - a. FE El usuario cancela la consulta.
 - b. Finaliza el caso de uso.
4. El sistema carga productos predefinidos existentes en el sistema según la categoría seleccionada.
5. El sistema carga detalle y características de productos disponibles en el sistema de catálogo.
 - a. FA No existen productos que cumplan con los criterios de búsqueda
 - b. El sistema despliega un mensaje que avisa sobre la inexistencia de productos que cumplan los criterios de búsqueda.
 - c. El usuario responde SI.
 - 5.c.1. FE El usuario no desea realizar otra búsqueda
 - 5.c.2. Finaliza el caso de uso.
 - d. El sistema evalúa la respuesta.
 - e. El sistema sugiere volver al punto 3 de flujo básico.
6. El usuario selecciona un producto.
7. El usuario escoge consultar detalle de producto.
8. El usuario visualiza las características completas del producto seleccionado.
9. Termina el caso de uso.

Requerimientos Especiales

No existen requerimientos especiales.

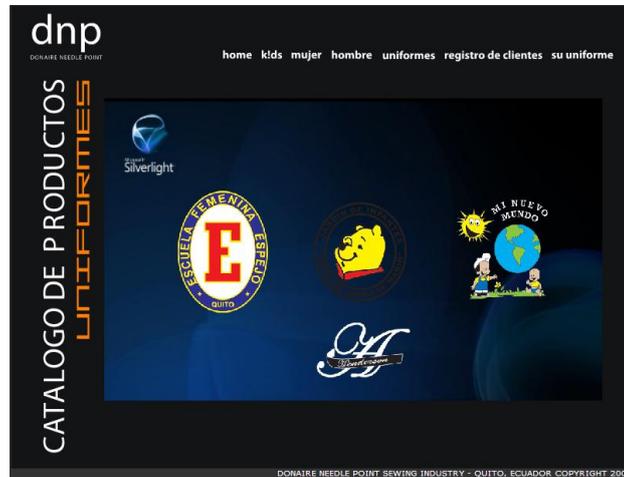
Diagrama

Outline Consultar Catálogo



Interfaz gráfica con el usuario

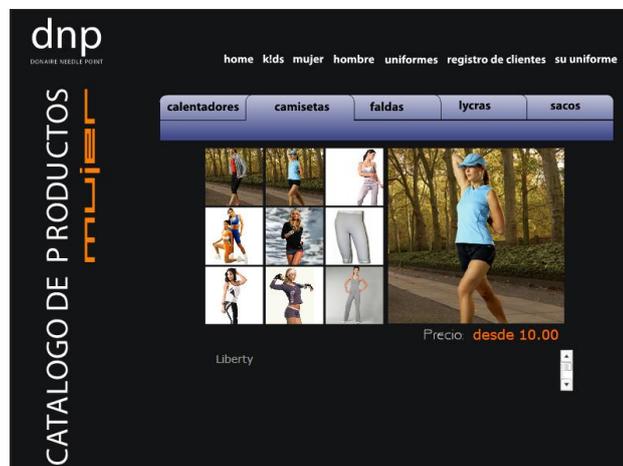
Pantalla de Consulta a catálogo 1



Pantalla de Consulta a catálogo 2



Pantalla de Consulta a catálogo 3



Relaciones

Interfaces con otros sistemas

No existen interfaces con otros sistemas.

Inclusiones

No existen.

Extensiones

No existen extensiones.

Suposiciones y Dudas Pendientes

No existen.

7.2.3 Validar Usuario

Breve Descripción

Este caso de uso muestra la funcionalidad del sistema cuando se valida la existencia de un usuario en el registro de base de datos.

Precondiciones

Conectividad con la aplicación Catálogo

Tanto el servicio de red como la conexión con la base de datos deben estar activos.

Pos condiciones de éxito**Consulta Efectuada**

El código empleado para obtener los datos devuelve un resultado.

Pos condiciones de falla**Falla en la red**

Existe un error de conectividad de red. Por lo que el sistema no logra invocar al módulo de validación.

Falla en la conexión con la Base de datos

Existe un error de conectividad con la Base de Datos.

Actor Principal

Catálogo de Productos

Flujo de Eventos**Flujo Básico**

Cuando un Usuario solicita ingresar al sistema de cotizaciones; se invoca al módulo de "Gestión de cotizaciones"; el mismo que a su vez invoca al módulo de Validación de Usuario el cual se encarga de validar la existencia de dicho usuario, ejecutando las siguientes operaciones:

1. El sistema envía los datos ingresados por el usuario
2. El sistema recibe los datos de la aplicación.
3. El sistema valida los datos ingresados mediante la ejecución de una consulta a la base de datos del sistema.

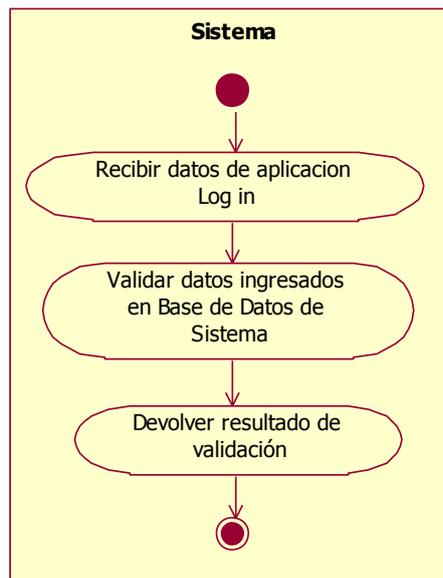
4. El sistema retorna el resultado de la validación.

Requerimientos Especiales

No existen requerimientos especiales.

Diagrama

Outline Validar Usuario



Pantalla de Interface

No existe.

Relaciones

Interfaces con otros sistemas

No existen interfaces con otros sistemas.

Inclusiones

Este caso de uso incluye el caso de uso "Gestionar Cotizaciones".

Extensiones

No existen extensiones.

7.2.4 Gestionar Imágenes

Breve Descripción

Este caso de uso muestra la funcionalidad del sistema cuando el Webmaster encargado del mantenimiento del sitio; efectúa cambios en lo referente a inserción, eliminación y actualización de imágenes del catálogo. Cabe mencionar que este proceso depende en gran parte del criterio y la intervención directa del Webmaster.

Precondiciones

Existencia de Imágenes

Debe existir imágenes, tratadas y preparadas para subirlas al sistema.

Existencia de Story Board Funcional y Mapa de Sitio

Las imágenes a ingresar deben seguir los parámetros definidos por la estructura y localidad definidas en el Mapa del Sitio y el Story Board Funcional.

Autorización de cambios por parte de la Empresa

Debe existir una autorización por escrito por parte de la empresa para efectuar cualquier cambio.

Pos condiciones de éxito**Imágenes cargadas exitosamente**

Se cargan las imágenes en la página Web y se guardan en el sistema.

Imágenes borradas exitosamente

Las imágenes seleccionadas por el Webmaster se eliminan del registro.

Imágenes reemplazadas exitosamente

Las imágenes seleccionadas por el usuario para modificación o reemplazo son guardadas con los cambios respectivos en el sistema.

Imágenes disponibles en catálogo

Las imágenes están disponibles para ser consultadas mediante la aplicación de catálogo.

Pos condiciones de falla**No se pueden mostrar las imágenes**

El sistema no es capaz de desplegar las imágenes debido a una falla de conectividad.

No se registraron las imágenes debido a un error en la base de datos del sistema

No se almacenaron las imágenes.

Imágenes mal ingresadas

Las imágenes ingresadas poseen incoherencia o defecto.

Actor Principal

Webmaster

Flujo de Eventos

Flujo Básico

El caso de uso inicia cuando el Webmaster solicita realizar las siguientes operaciones de gestión de imágenes:

- SF Webmaster Solicita ingresar imagen a catálogo.
- SF Webmaster solicita borrar imagen seleccionada
- SF Webmaster solicita modificar la imagen seleccionada
- Termina el caso de uso.

Sub flujos

Webmaster solicita ingresar imagen

Si el Webmaster solicitó ingresar imagen, se ejecutarán las siguientes operaciones:

1. El Webmaster selecciona la imagen o imágenes a ingresar.
2. El Webmaster efectúa el respectivo tratamiento de ajuste de calidad de imagen, además de modificar el tamaño y forma de las mismas conforme se requiera.
3. El Webmaster clasifica la o las imágenes a ingresar al catálogo
4. El Webmaster carga las imágenes al sistema
5. El sistema almacena las imágenes insertadas

Webmaster solicita borrar imagen seleccionada

Si el Webmaster decide eliminar una imagen se ejecutarán las siguientes operaciones:

1. El Webmaster busca la imagen a eliminar.
2. El sistema muestra la colección de imágenes.
3. El Webmaster selecciona la imagen a eliminar.
4. El sistema elimina la imagen.
5. El sistema guarda los cambios

Webmaster solicita modificar la imagen seleccionada

El Webmaster tendrá que hacer modificaciones al contenido de imágenes existentes en el sistema. Estas modificaciones contemplan:

- Cambio de contenido; para el caso de insertar o quitar elementos gráficos a una imagen existente.
- Actualización, para el caso en el que se haya cambiado algún elemento de la imagen.
- Corrección; en el caso de haber defectos.
- Reemplazo de ser necesario cambiar la imagen en su totalidad

Para lo cual se seguirá el siguiente proceso

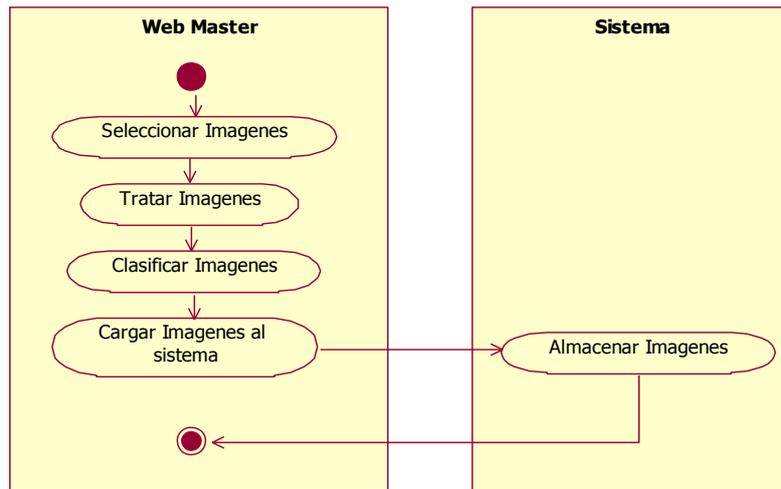
1. El Webmaster busca la imagen a modificar
2. El sistema muestra la colección de imágenes según categoría.
3. El Webmaster selecciona la imagen a modificar.
4. El sistema reemplaza la imagen existente por la imagen actualizada.

Requerimientos Especiales

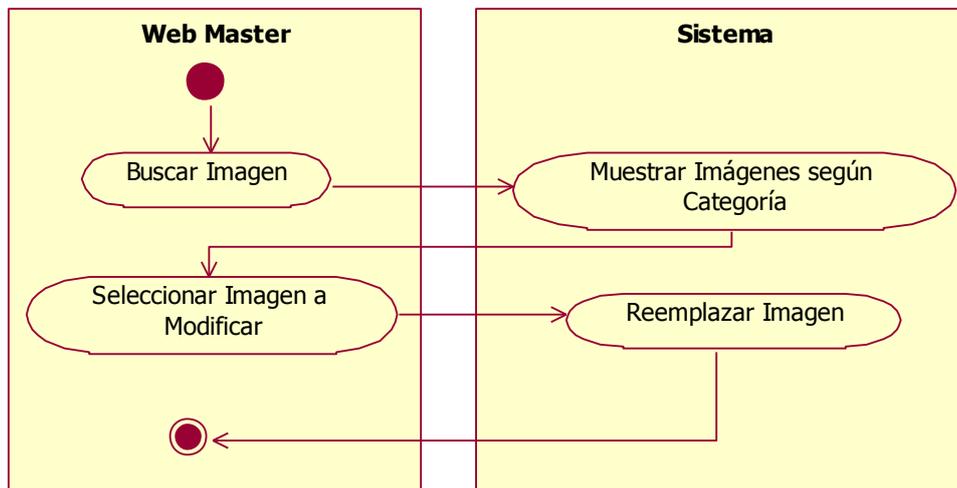
No existen requerimientos especiales.

Diagramas

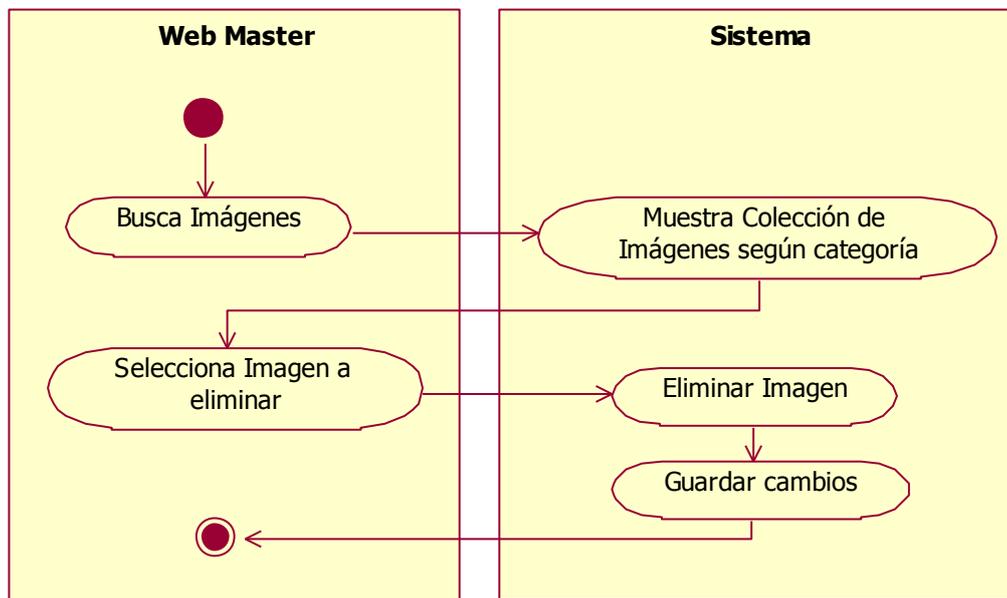
Outline Ingresar Imagen



Outline Modificar Imagen



Outline Eliminar Imagen



Relaciones

Interfaces con otros sistemas

No existen interfaces con otros sistemas.

Inclusiones

Este caso de uso incluye al módulo “Catálogo de Productos”

Extensiones

No existen extensiones.

7.2.5 Gestionar Categorías

Breve Descripción

Este caso de uso muestra la funcionalidad del sistema cuando el Webmaster encargado del mantenimiento del sitio; efectúa cambios en lo referente a inserción, eliminación y actualización de Categorías del catálogo. Cabe

mencionar que este proceso depende en gran parte del criterio y la intervención directa del Webmaster.

Precondiciones

Existencia de Categorías

Debe existir categorías corregidas y adecuadas para subirlas al sistema.

Existencia de Story Board Funcional y Mapa de Sitio

Las Categorías a ingresar deben seguir los parámetros definidos por la estructura y localidad definidas en el Mapa del Sitio y el Story Board Funcional.

Autorización de cambios por parte de la Empresa

Debe existir una autorización por escrito por parte de la empresa para efectuar cualquier cambio.

Pos condiciones de éxito

Categorías cargadas exitosamente

Se cargan las Categorías en la página Web y se guardan en el sistema.

Categorías borradas exitosamente

Las Categorías seleccionadas por el Webmaster se eliminan del registro.

Categorías reemplazadas exitosamente

Las Categorías seleccionadas por el usuario para modificación o reemplazo son guardadas con los cambios respectivos en el sistema.

Categorías disponibles en catálogo

Las imágenes están disponibles para ser consultadas mediante la aplicación de catálogo.

Pos condiciones de falla

No se pueden mostrar las Categorías

El sistema no es capaz de desplegar las Categorías debido a una falla de conectividad.

No se registraron las Categorías debido a un error en la base de datos del sistema

No se almacenaron las Categorías.

Categorías mal ingresadas

Las Categorías ingresadas poseen incoherencia o defecto.

Actor Principal

Webmaster

Flujo de Eventos

Flujo Básico

1. El caso de uso inicia cuando el Webmaster solicita realizar las siguientes operaciones de gestión de Categorías:
 - SF Webmaster Solicita ingresar Categorías a catálogo.
 - SF Webmaster solicita borrar Categorías seleccionada
 - SF Webmaster solicita modificar la Categorías seleccionada
2. Termina el caso de uso.

Sub flujo

Webmaster solicita ingresar imagen

Si el Webmaster solicitó ingresar categoría, se ejecutarán las siguientes operaciones

1. El Webmaster planifica la nueva categoría a ingresar según parámetros funcionales y estéticos de la aplicación.
2. El Webmaster define la nueva categoría.
3. El Sistema agrega la nueva categoría.
4. El sistema Guarda los cambios efectuados.

Usuario solicita borrar categoría seleccionada

Si el Webmaster decide eliminar una categoría se ejecutarán las siguientes operaciones:

1. El Webmaster busca la categoría a eliminar.
2. El sistema carga las categorías existentes.
3. El Webmaster selecciona la categoría a eliminar.
4. El sistema elimina la categoría.
5. El Sistema guarda los cambios.

Usuario solicita modificar la categoría seleccionada

El Webmaster hará modificaciones al contenido de las categorías existentes en el sistema. Para lo cual se seguirá el siguiente proceso

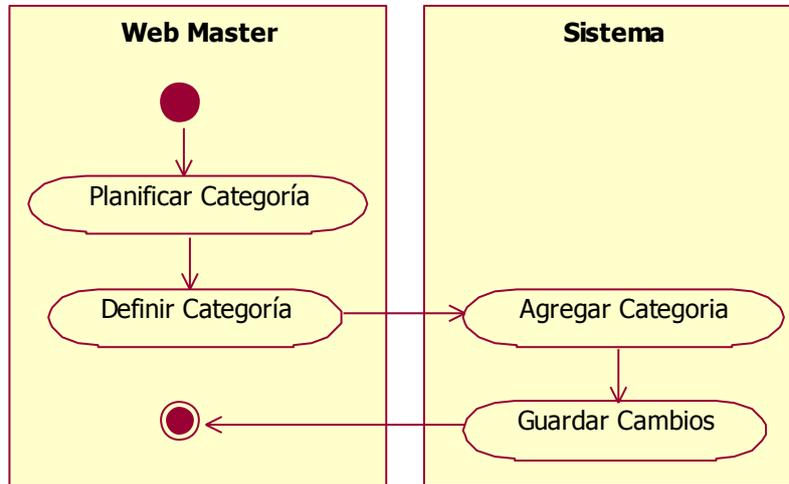
1. El Webmaster busca la categoría a modificar
2. El sistema carga las categorías existentes.
3. El Webmaster selecciona la categoría a modificar.
4. El Webmaster Modifica la categoría.
5. El sistema guarda los cambios efectuados sobre la categoría

Requerimientos Especiales

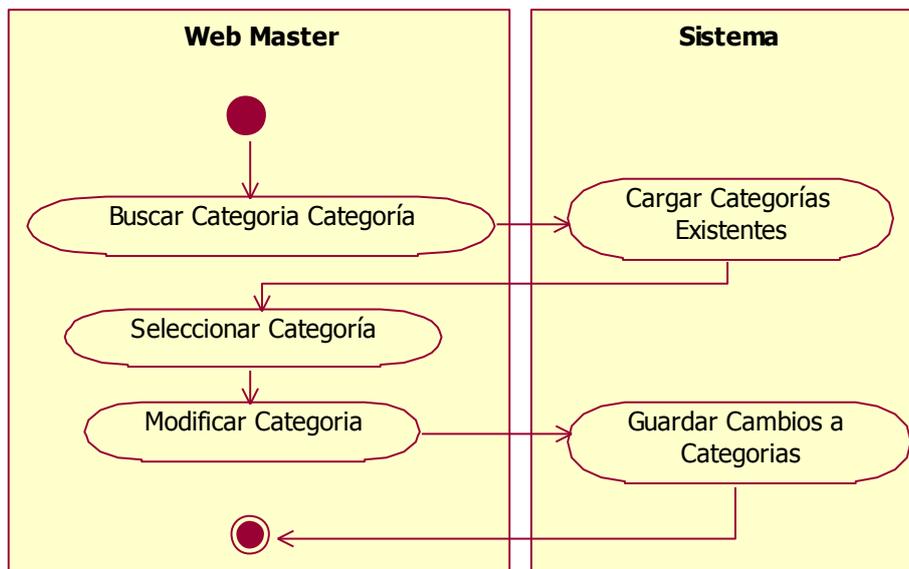
No existen requerimientos especiales.

Diagramas

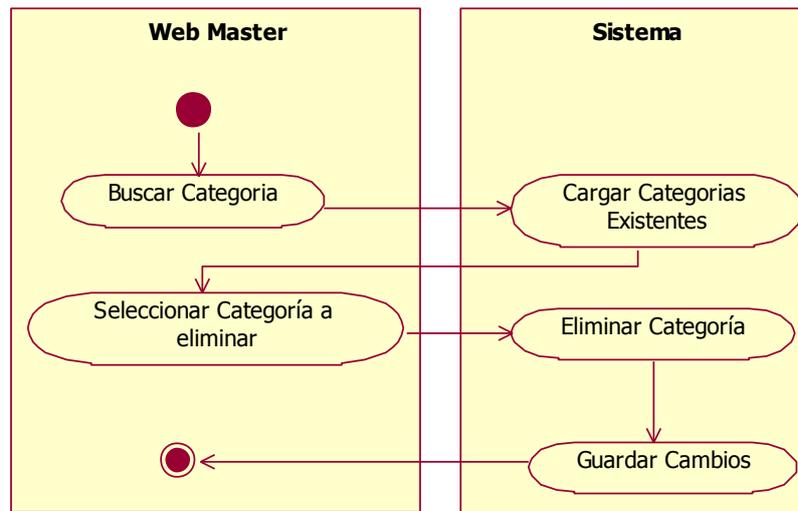
Outline Ingresar Categoría



Outline Modificar Categoría

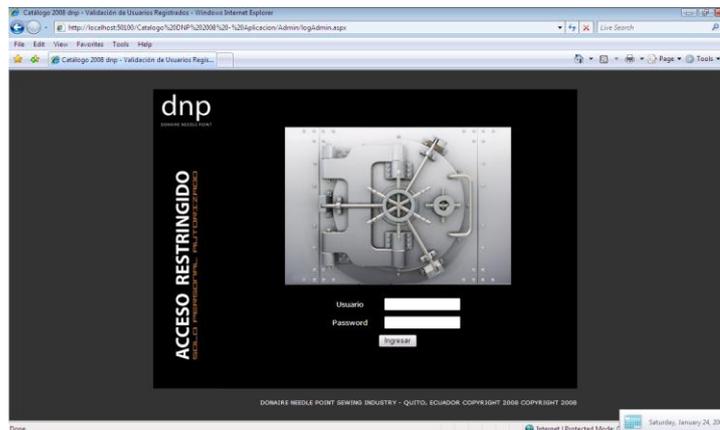


Outline Eliminar Categoría



Interfaz gráfica con el usuario

Pantalla 1 de Ingreso a Modulo Administrativo



Pantalla 2 de Menú para Modulo Administrativo



Pantalla 3 de Ingreso de Categoría – Extras

The screenshot shows the 'Agregar' (Add) screen in the 'Administrador' interface. The page title is 'Administrador' and the breadcrumb is 'Agregar'. The main content area contains a form with the following fields:

- Nombre Extra:
- Tamano:
- Comp.Cromatica:
- Valor:

There is an 'Agregar' button at the bottom of the form. The left sidebar contains a menu with 'Extras' selected. The footer text reads 'DONAIRE NEEDLE POINT SEWING INDUSTRY - QUITO, ECUADOR COPYRIGHT 2008'.

Pantalla 4 de Modificación de Categoría – Extras

The screenshot shows the 'Modificar' (Modify) screen in the 'Administrador' interface. The page title is 'Administrador' and the breadcrumb is 'Modificar'. The main content area contains a form with the following fields:

- Id Extra:
- Nombre Extra:
- Tamano:
- Comp.Cromatica:
- Valor:

There is a 'Modificar' button at the bottom of the form. The left sidebar contains a menu with 'Extras' selected. The footer text reads 'DONAIRE NEEDLE POINT SEWING INDUSTRY - QUITO, ECUADOR COPYRIGHT 2008'.

Pantalla 5 para Eliminar Categoría – Extras

The screenshot shows the 'Eliminar' (Delete) screen in the 'Administrador' interface. The page title is 'Administrador' and the breadcrumb is 'Eliminar'. The main content area contains a form with the following fields:

- Id Extra:

There is an 'Eliminar' button at the bottom of the form. The left sidebar contains a menu with 'Extras' selected. The footer text reads 'DONAIRE NEEDLE POINT SEWING INDUSTRY - QUITO, ECUADOR COPYRIGHT 2008'.

Pantalla 6 de Ingreso de Categoría – Telas

Administrador

Consultar • Modificar • Insertar • Eliminar

Nombre

Tipo Calentador

Valor

Insertar

DONAIRE NEEDLE POINT SEWING INDUSTRY - QUITO, ECUADOR COPYRIGHT 2008

Pantalla 7 de Modificación de Categoría – Telas

Administrador

Consultar • Modificar • Insertar • Eliminar

ID Tela 2

Nombre

Tipo Calentador

Valor

Modificar

DONAIRE NEEDLE POINT SEWING INDUSTRY - QUITO, ECUADOR COPYRIGHT 2008

Pantalla 8 para Eliminar Categoría – Telas

Administrador

Consultar • Modificar • Insertar • Eliminar

Nombre Topper

Eliminar

DONAIRE NEEDLE POINT SEWING INDUSTRY - QUITO, ECUADOR COPYRIGHT 2008

Pantalla 9 de Ingreso de Categoría – Colores



Pantalla 10 de Modificación de Categoría – Colores



Pantalla 11 para Eliminar Categoría – Colores



Relaciones

Interfaces con otros sistemas

No existen interfaces con otros sistemas.

Inclusiones

Este caso de uso incluye al módulo “Catálogo de Productos”

Extensiones

No existen extensiones.

7.2.6 Gestionar Productos

Breve Descripción

Este caso de uso muestra la funcionalidad del sistema cuando el Webmaster encargado del mantenimiento del sitio; efectúa cambios en lo referente a inserción, eliminación y actualización de Productos del catálogo. Cabe mencionar que este proceso depende en gran parte del criterio y la intervención directa del Webmaster.

Precondiciones

Existencia de Productos

Debe existir Productos disponibles para ofertarlos mediante el sistema.

Pos condiciones de éxito

Productos cargados exitosamente

Se cargan las Productos en la página Web y se guarda el detalla de producto el sistema.

Productos eliminados exitosamente

Los productos seleccionados por el Webmaster se eliminan del registro.

Productos reemplazados exitosamente

Los productos seleccionados por el usuario para modificación o reemplazo son guardados con los cambios respectivos en la base de datos.

Productos disponibles en catálogo

Los productos están disponibles para ser consultados mediante la aplicación de catálogo.

Pos condiciones de falla**No se pueden mostrar los productos**

El sistema no es capaz de desplegar los productos debido a una falla de conectividad o una falla interna del sistema de consultas.

No se registraron los productos debido a un error en la base de datos del sistema

No se almacenaron los productos.

Productos mal ingresados

Los productos ingresados contienen errores en su detalle de información.

Actor Principal

Webmaster

Flujo de Eventos

Flujo Básico

- El caso de uso inicia cuando el Webmaster solicita realizar las siguientes operaciones de gestión de productos:
- SF Webmaster Solicita ingresar productos a catálogo.
- SF Webmaster solicita borrar productos seleccionados
- SF Webmaster solicita modificar un producto seleccionado
- Termina el caso de uso.

Sub flujo

Webmaster solicita ingresar producto

Si el Webmaster solicitó ingresar producto, se ejecutarán las siguientes operaciones:

1. El Webmaster selecciona el producto a ingresar al catálogo.
2. El sistema consulta las categorías disponibles.
3. EL Webmaster clasifica el producto a ingresar en una de las categorías.
4. El Webmaster ingresa el producto.
5. El sistema almacena el producto.
6. El Webmaster detalla el producto.
7. El sistema almacena el detalle de producto.
8. El sistema guarda los cambios efectuados.

Usuario solicita borrar producto

Si el Webmaster decide eliminar un producto se ejecutarán las siguientes operaciones:

- El Webmaster busca el producto a eliminar.
- El sistema muestra los productos del catálogo.
- El Webmaster selecciona el producto a eliminar.
- El sistema elimina el producto seleccionado.
- El sistema guarda los cambios.

Usuario solicita modificar producto

Si el Webmaster requiere hacer modificaciones al detalle de los productos existentes:

- Cambio de información relativa al producto.
- Reemplazo de ser necesario cambiar el contenido de producto en general.

Para lo cual se seguirá el siguiente proceso

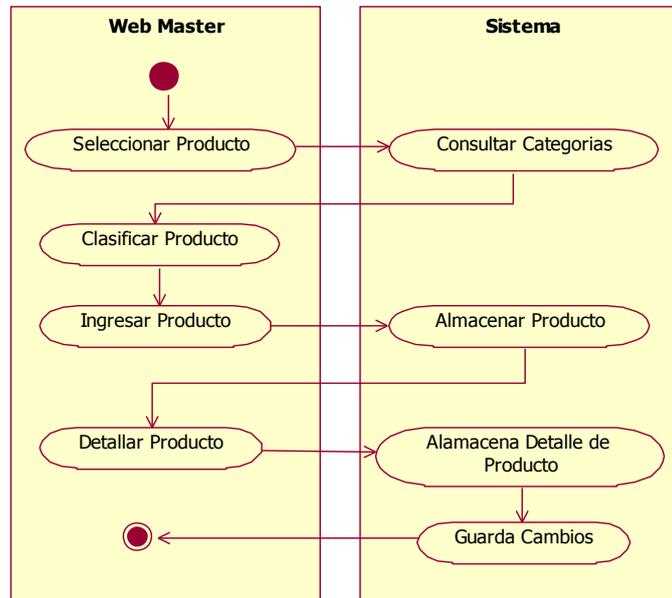
1. El Webmaster busca el producto a modificar
2. El sistema muestra los productos según categoría.
3. El Webmaster selecciona el producto a modificar.
4. El Webmaster realiza las modificaciones necesarias.
5. El sistema guarda los cambios realizados.

Requerimientos Especiales

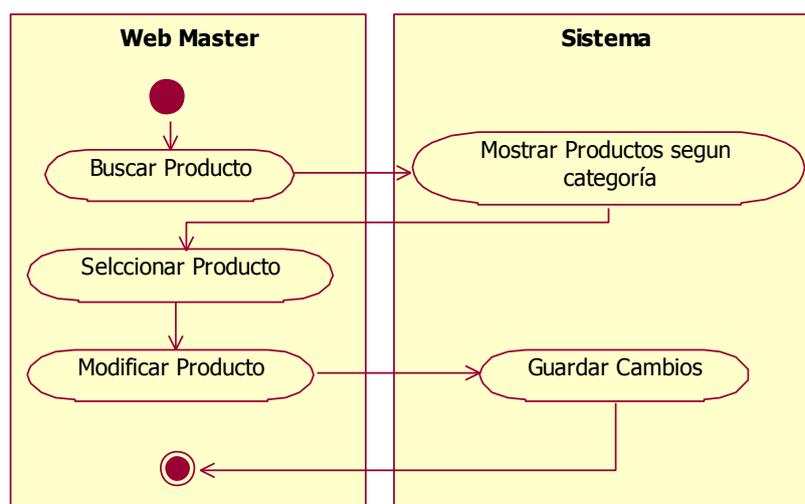
No existen requerimientos especiales.

Diagramas

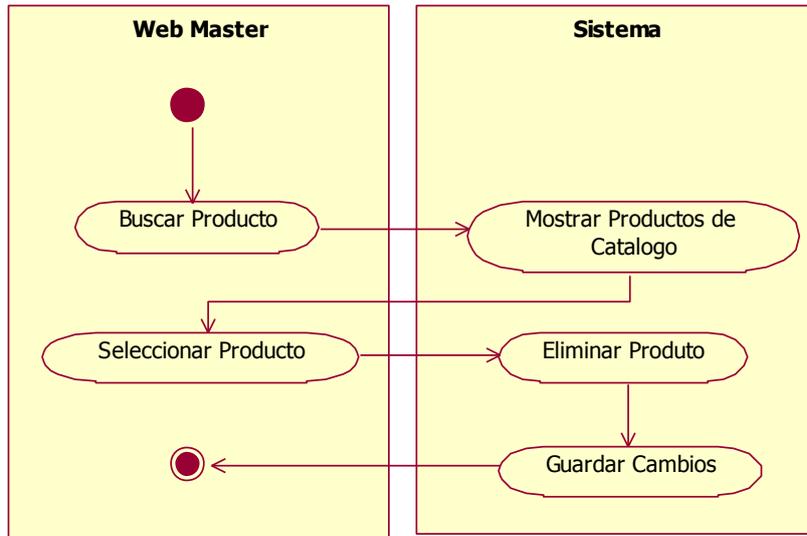
Outline Ingresar Producto



Outline Modificar Producto

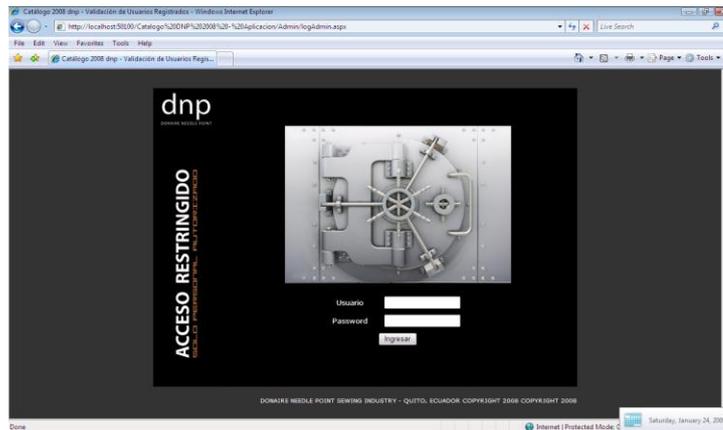


Outline Eliminar Producto



Interfaz gráfica con el usuario

Pantalla 1 de Ingreso a Modulo Administrativo



Pantalla 2 de Menú para Modulo Administrativo



Pantalla 3 para selección de menú de ingreso



Pantalla 4 de Ingreso de Producto a Catalogo



Pantalla 5 de Ingreso de Producto como Producto



Pantalla 6 de modificación de Producto a Catalogo



Pantalla 7 de modificación de Producto como Producto



Pantalla 8 para eliminar Producto



Relaciones

Interfaces con otros sistemas

No existen interfaces con otros sistemas.

Inclusiones

Este caso de uso incluye al módulo “Catálogo de Productos”

Extensiones

No existen extensiones.

7.2.7 Gestionar Novedades

Breve Descripción

Este caso de uso muestra la funcionalidad del sistema cuando el Webmaster encargado del mantenimiento del sitio; efectúa cambios en lo referente a inserción, eliminación y actualización de novedades y artículos de interés como extras del catálogo de productos. Cabe mencionar que este proceso depende en gran parte del criterio y la intervención directa del Webmaster.

Precondiciones

Existencia de Novedades

Deben existir artículos y noticias relativos a prendas de vestir; revisados y listos para subirlos al sistema.

Existencia de Story Board Funcional y Mapa de Sitio

Las novedades y artículos deberán ser ingresados siguiendo los parámetros definidos por la estructura y localidad definidas en el Mapa del Sitio y el Story Board Funcional.

Autorización de cambios por parte de la Empresa

Debe existir una autorización por escrito por parte de la empresa para efectuar cualquier cambio.

Pos condiciones de éxito**Novedades cargadas exitosamente**

Se cargan las novedades en la página Web y se guardan en el sistema.

Novedades borradas exitosamente

Las novedades seleccionadas por el Webmaster se eliminan del registro.

Novedades reemplazadas exitosamente

Las novedades seleccionadas por el Webmaster para modificación o reemplazo son guardadas con los cambios respectivos en el sistema.

Novedades disponibles en catálogo

Las novedades están disponibles para ser consultadas mediante la aplicación de catálogo.

Pos condiciones de falla**No se pueden mostrar las novedades**

El sistema no es capaz de desplegar las novedades debido a una falla de conectividad.

No se registraron las novedades debido a un error en el archivo XML de novedades.

No se almacenaron las novedades.

Novedades mal ingresadas

Las novedades ingresadas poseen incoherencia o defecto.

Actor Principal

Webmaster

Flujo de Eventos

Flujo Básico

1. El caso de uso inicia cuando el Webmaster solicita realizar las siguientes operaciones de gestión de imágenes:
 - SF Webmaster Solicita ingresar novedades a catálogo.
 - SF Webmaster solicita borrar novedad seleccionada
 - SF Webmaster solicita modificar la novedad seleccionada
2. Termina el caso de uso.

Sub flujo

Webmaster solicita ingresar novedad

Si el Webmaster solicitó ingresar novedad, se ejecutarán las siguientes operaciones:

1. El Webmaster clasifica la novedad o artículo a ingresar.
2. El Webmaster, crea el archivo XML para el nuevo artículo o novedad.
3. EL Webmaster ingresa la novedad al archivo.
4. El sistema almacena el archivo creado.

Usuario solicita borrar novedad seleccionada

Si el Webmaster decide eliminar novedad se ejecutarán las siguientes operaciones:

1. El Webmaster busca la novedad a eliminar.
2. El sistema muestra los archivos de novedades disponibles.
3. El Webmaster selecciona el archivo XML de novedad a eliminar.
4. El Webmaster elimina el archivo.

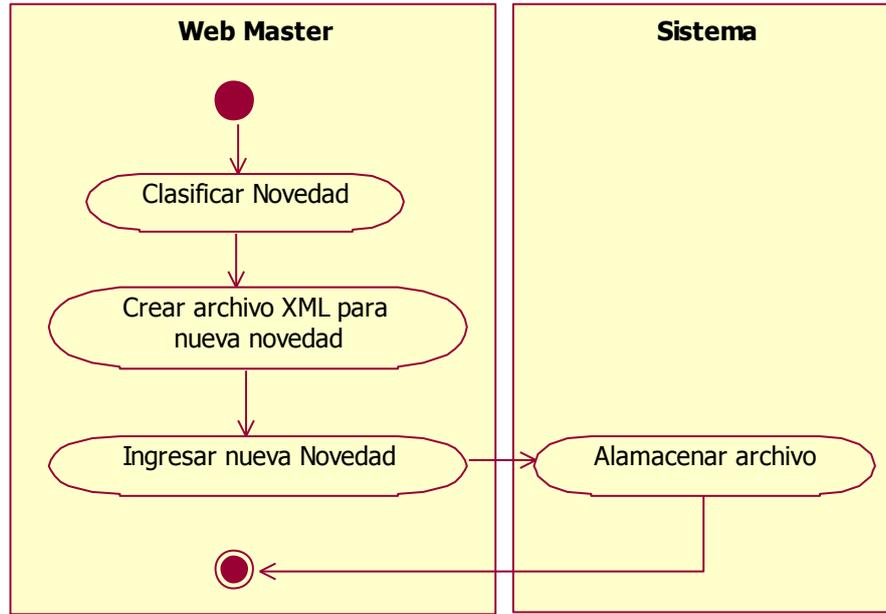
Usuario solicita modificar novedad seleccionada

El Webmaster tendrá que hacer modificaciones al contenido de novedades y artículos existentes en el sistema. Estas modificaciones contemplan, todo cambio de contenido; para los casos de corrección actualización o reemplazo. Para lo cual se seguirá el siguiente proceso

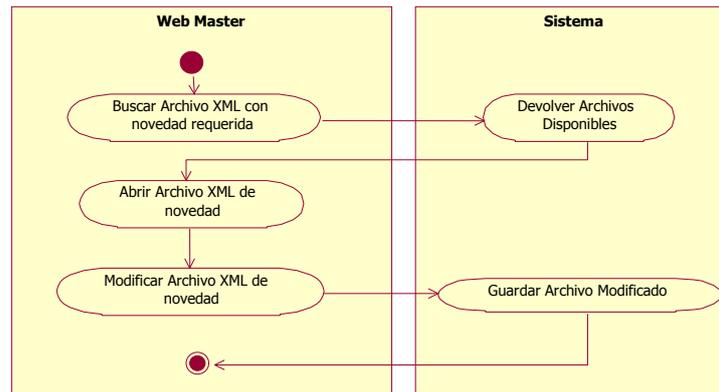
1. El Webmaster busca el archivo que contiene la novedad a modificar
2. El sistema muestra la colección de archivos disponibles.
3. El Webmaster abre el archivo a modificar.
4. El Webmaster modifica la novedad o artículo del archivo abierto.
5. El sistema guarda los cambios del archivo modificado.

Diagramas

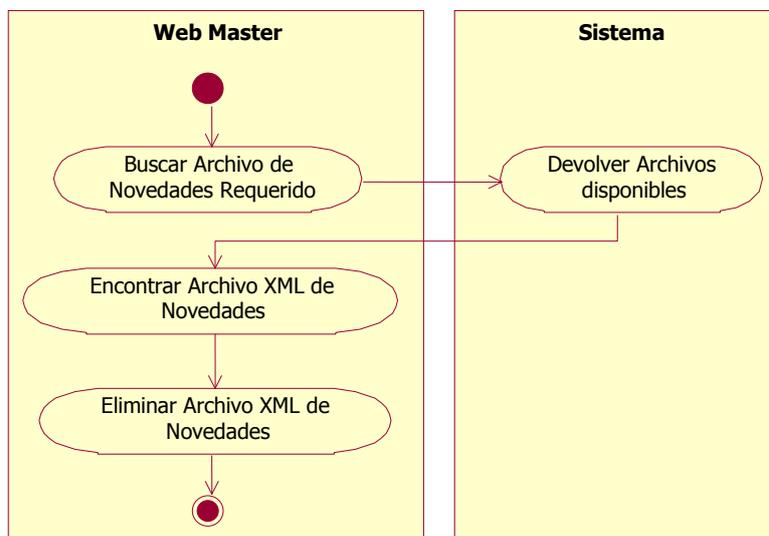
Outline Ingresar Novedad



Outline Modificar Novedad



Outline Eliminar Novedad



Relaciones

Interfaces con otros sistemas

No existen interfaces con otros sistemas.

Inclusiones

Este caso de uso incluye al módulo “Catálogo de Productos”

Extensiones

No existen extensiones.

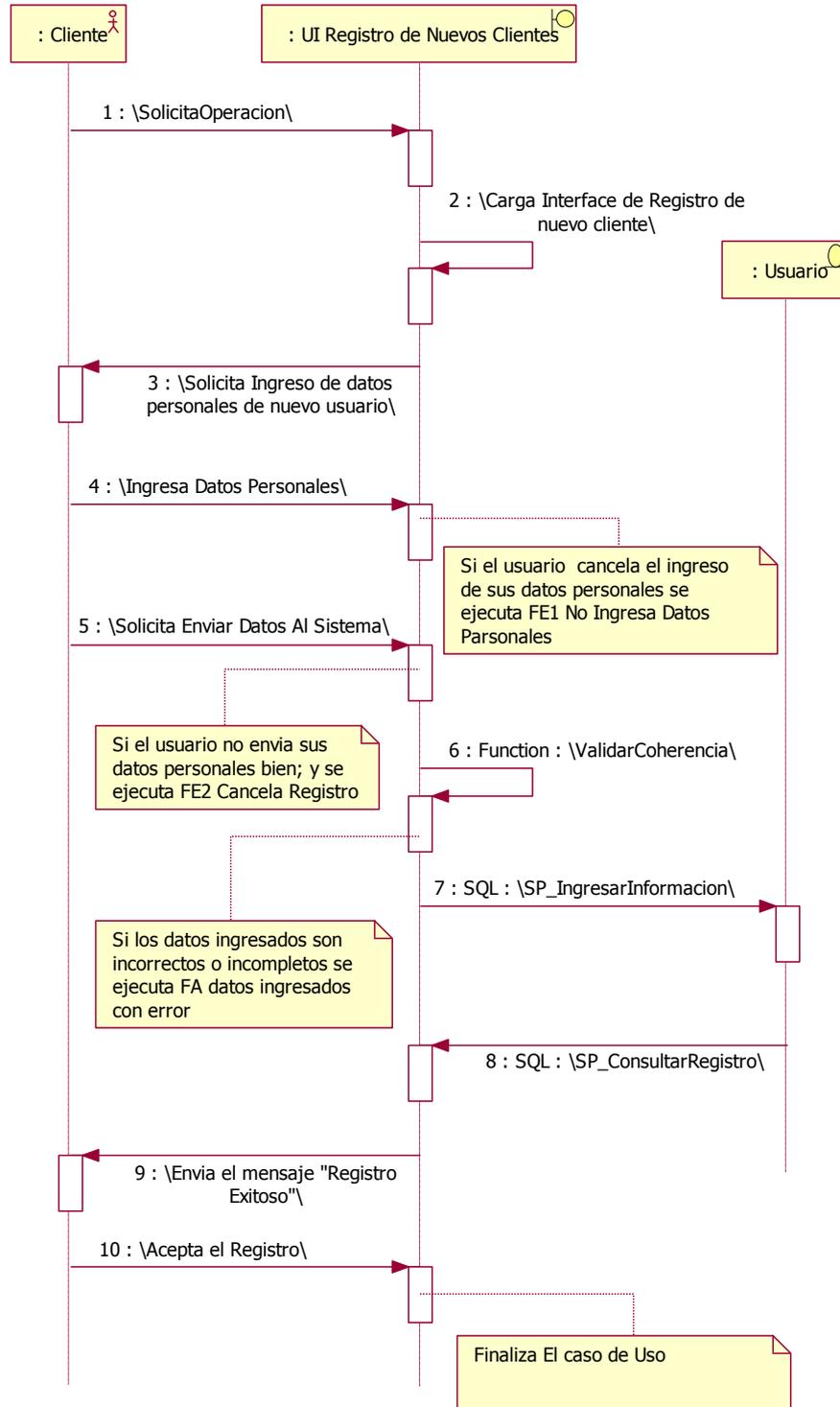
7.3. Anexo “C”: Diagramas De Secuencia Secundarios

DIAGRAMAS DE SECUENCIA

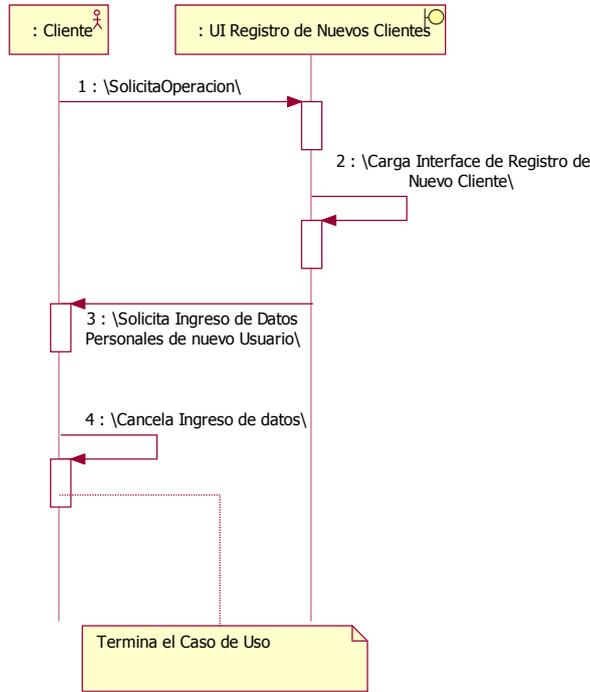
“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

7.3.1 Registrar Usuario

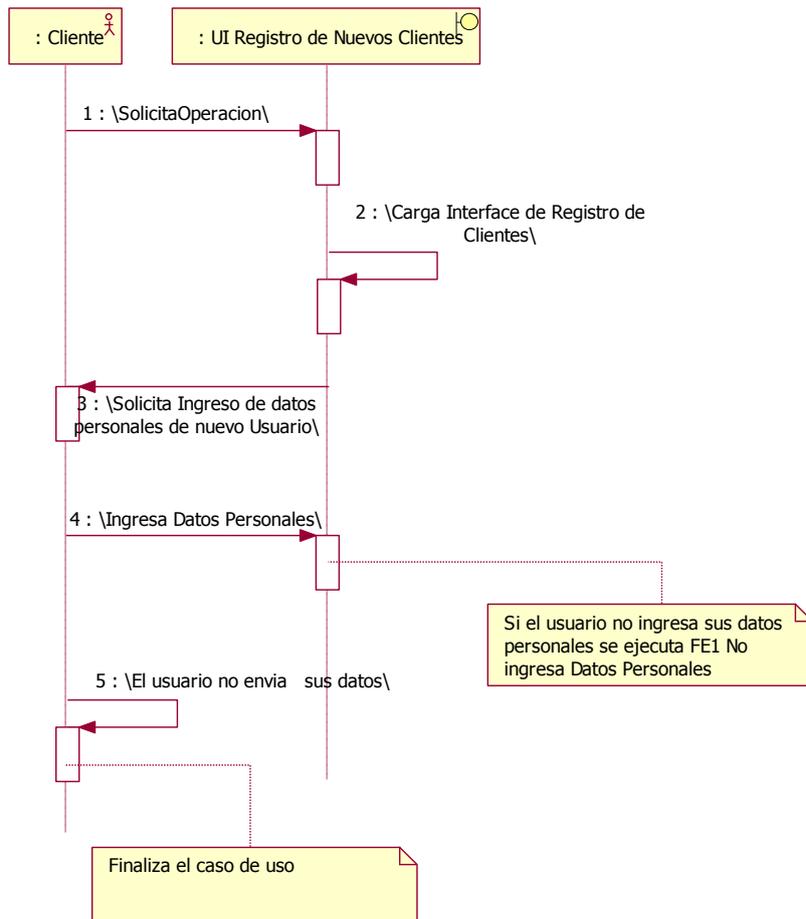
Flujo Básico: Registrar Usuario



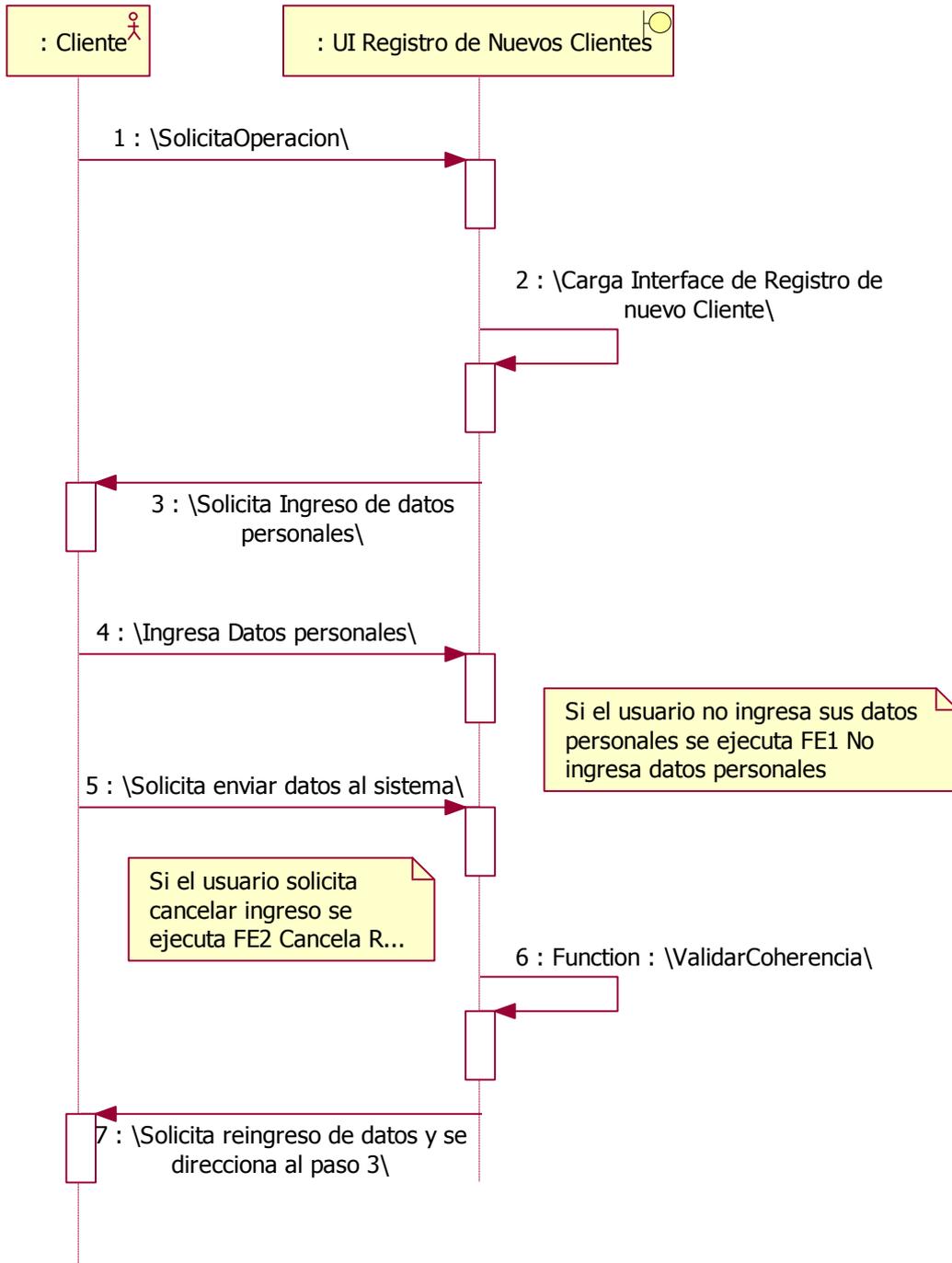
Flujo de Excepción: FE1 No Ingresa datos Personales



Flujo de Excepción: FE2 No envía correctamente sus datos personales

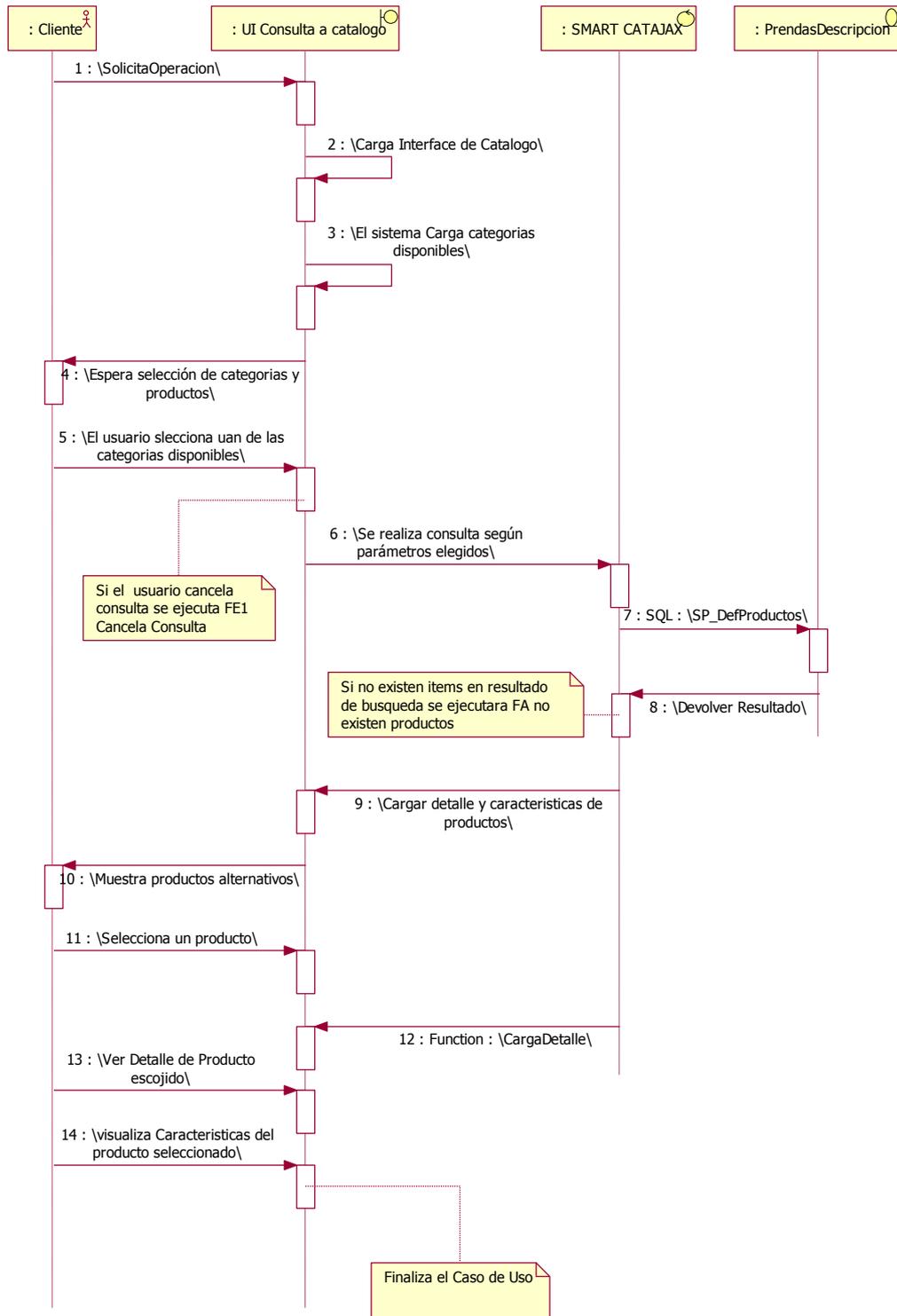


Flujo de Excepción: FA Datos Ingresados con error

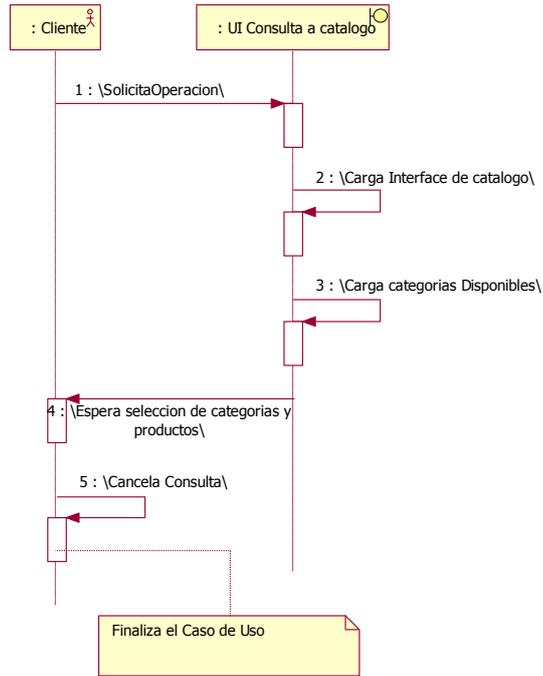


7.3.2 Consultar Catálogo

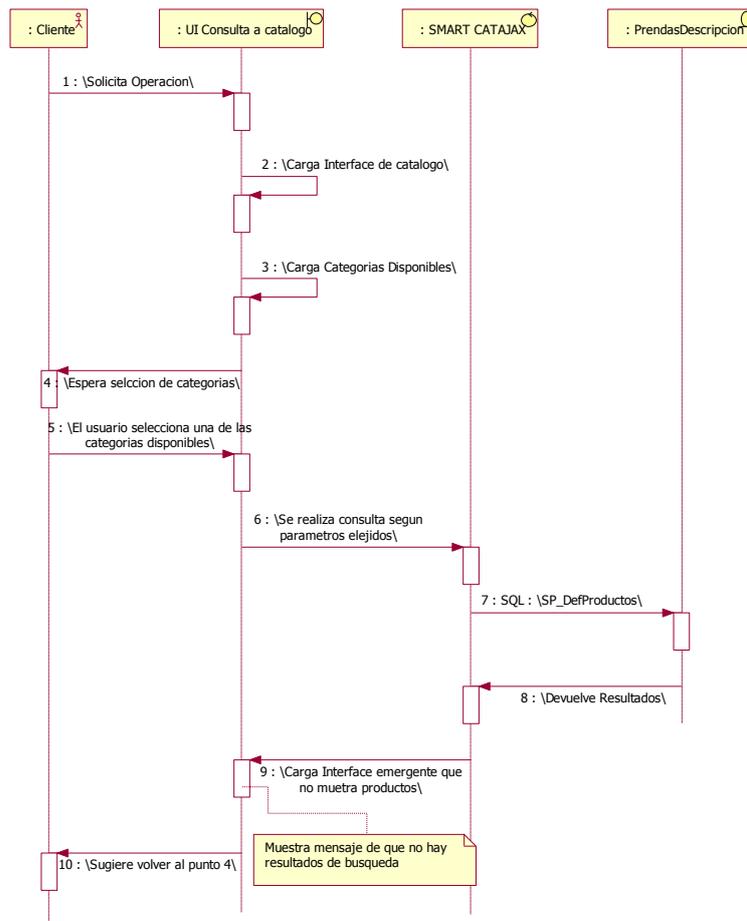
Flujo Básico: Consultar Catálogo



Flujo de Excepción: FE Cancela Consulta

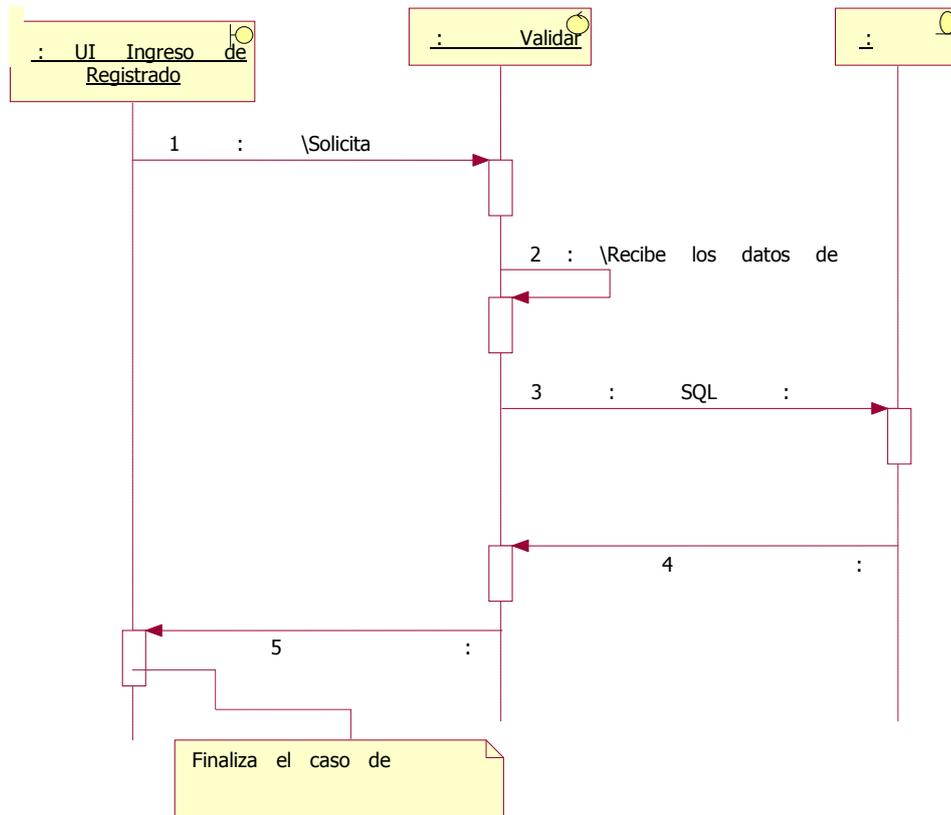


Flujo Alternativo: FA No existen Productos



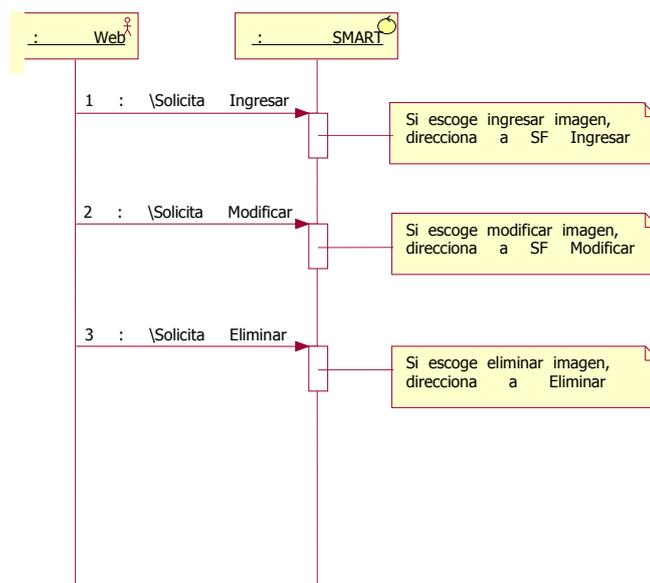
7.3.3 Validar Usuario

Flujo Básico: Registrar Usuario

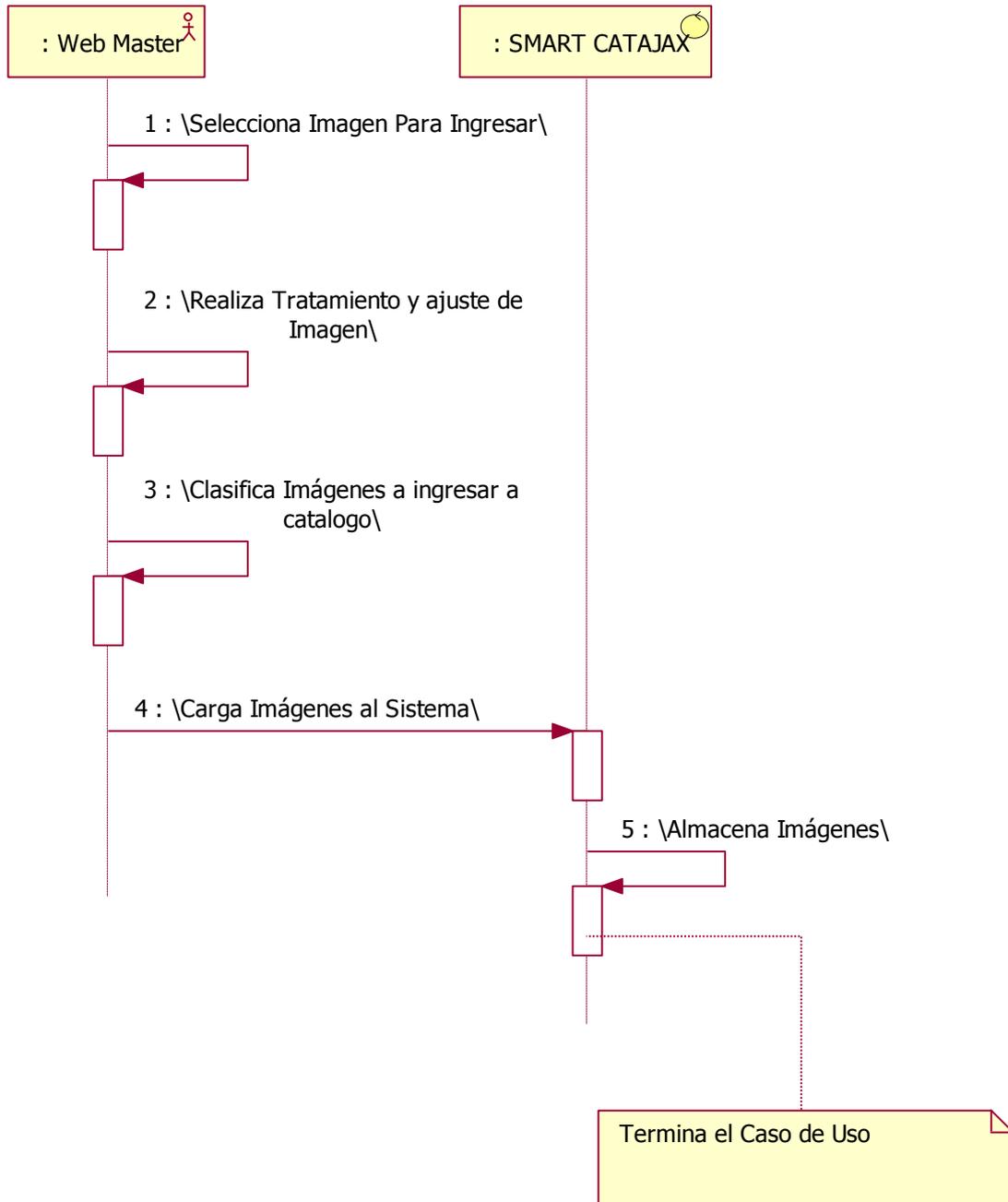


7.3.4 Gestionar Imágenes

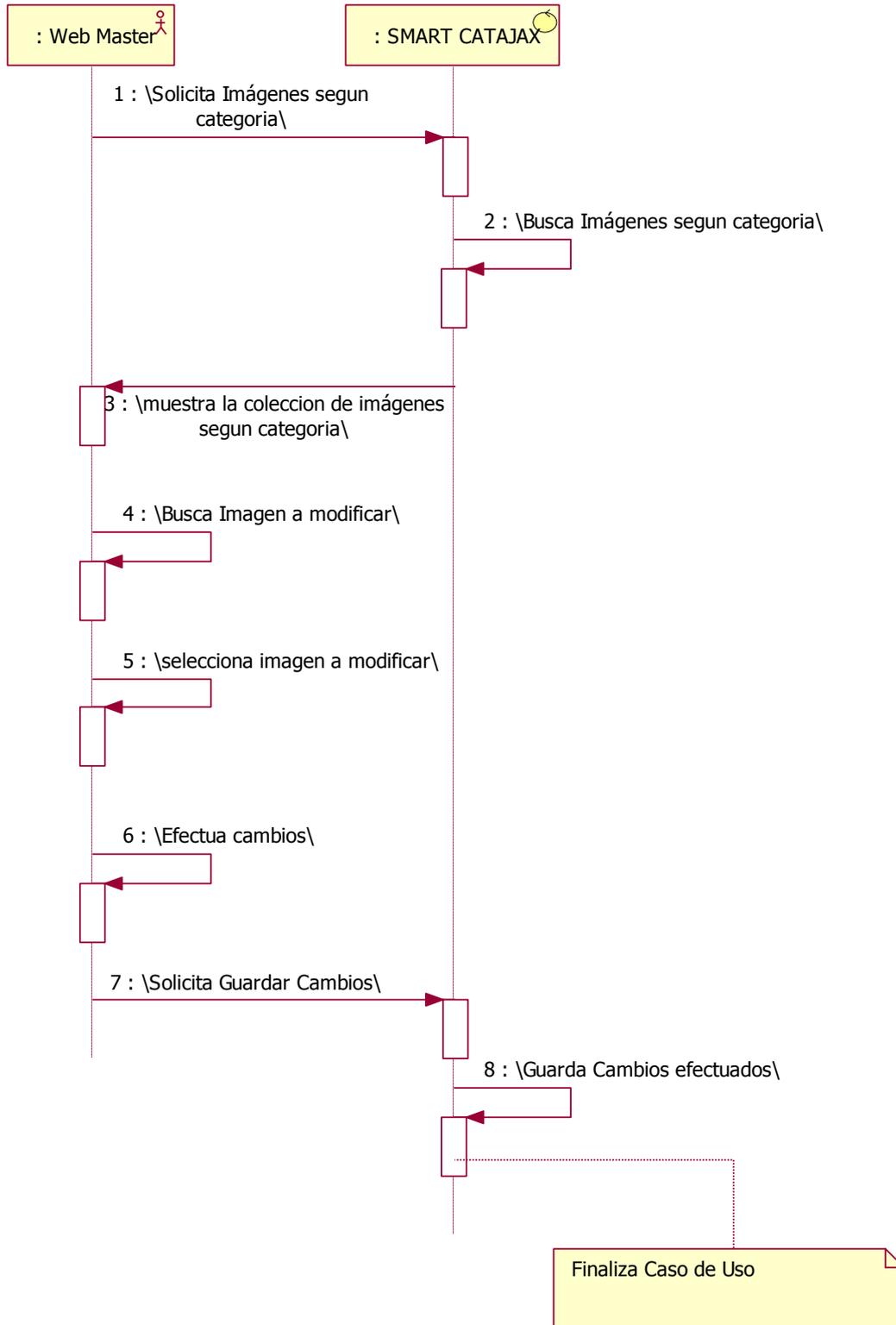
Flujo Básico: Gestionar Imágenes



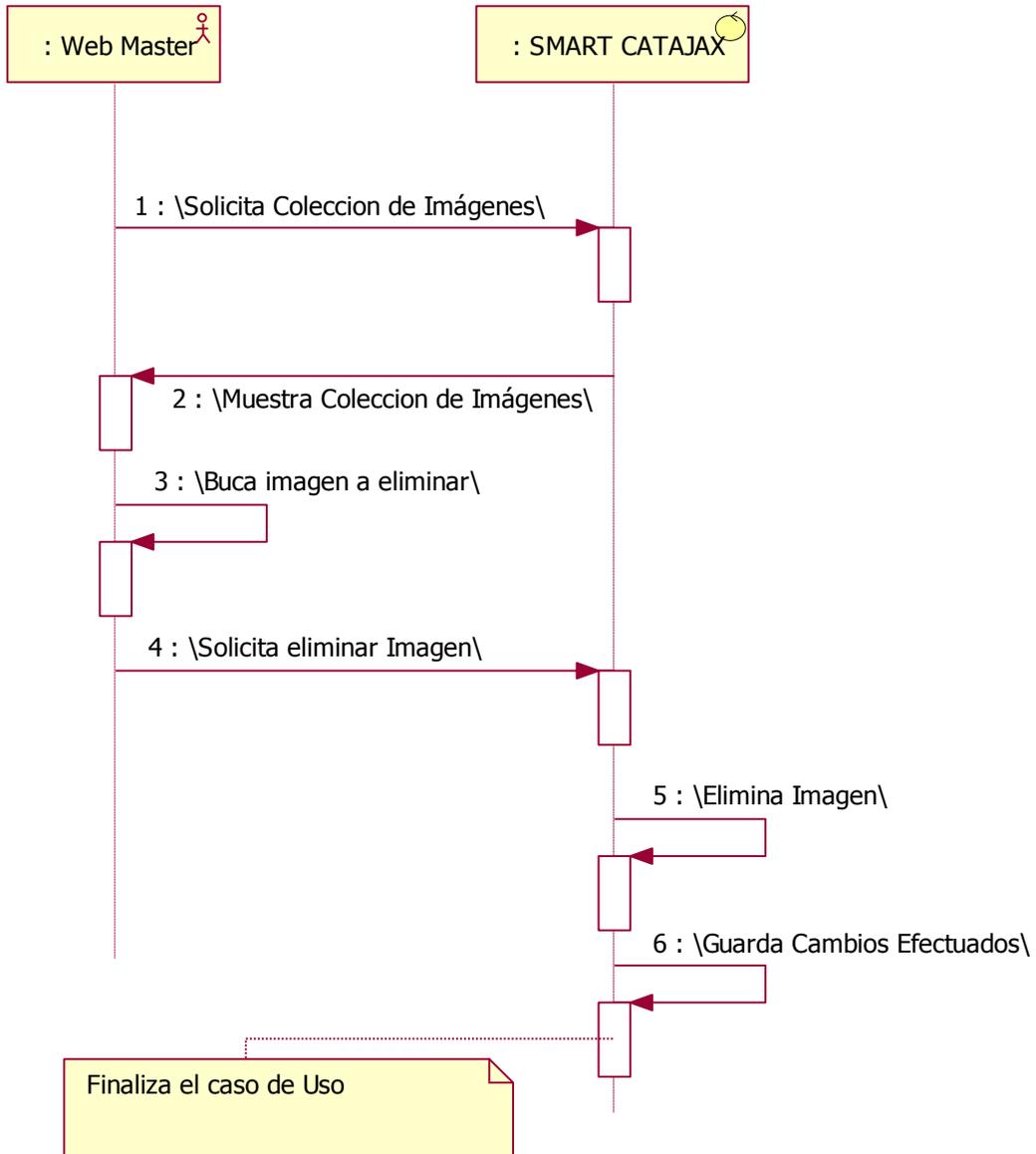
Sub flujo: Insertar Imágenes



Sub flujo: Modificar Imágenes

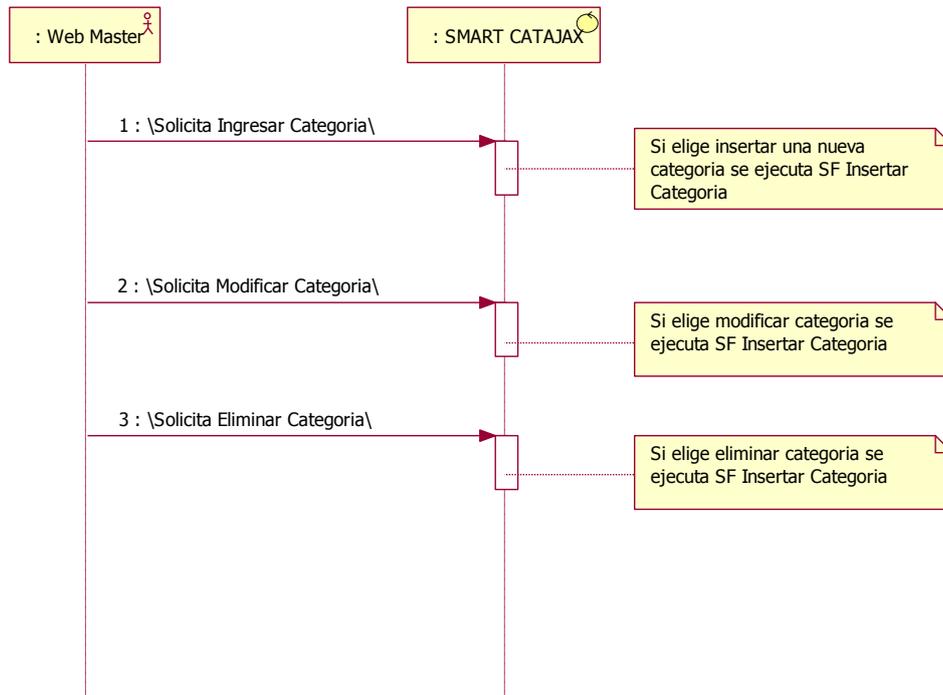


Sub flujo: Eliminar Imágenes

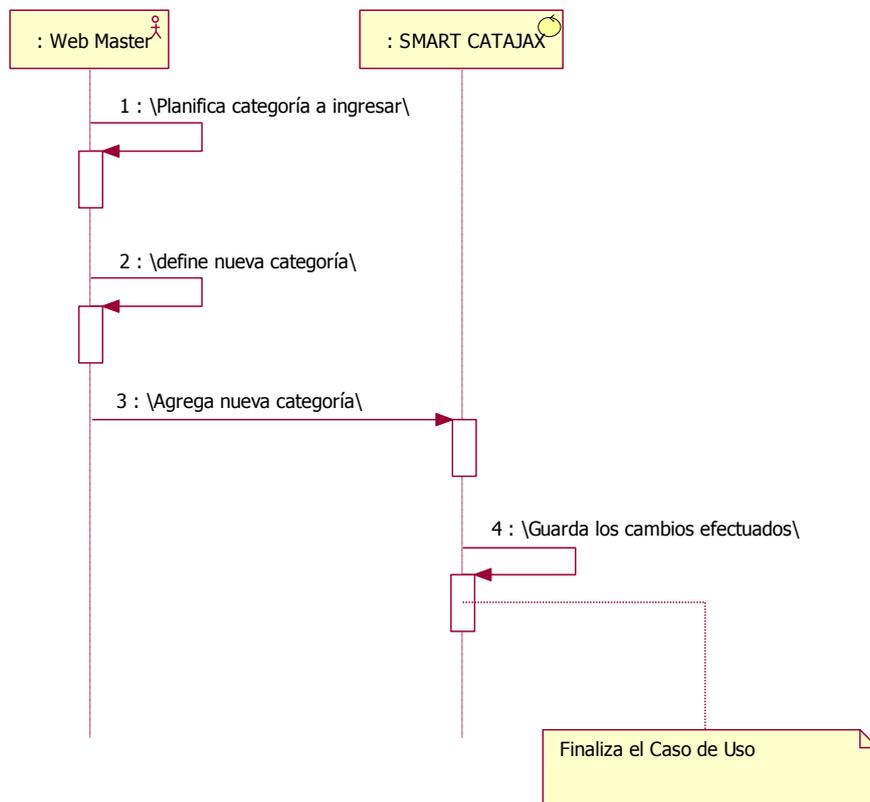


7.3.5 Gestionar Categorías

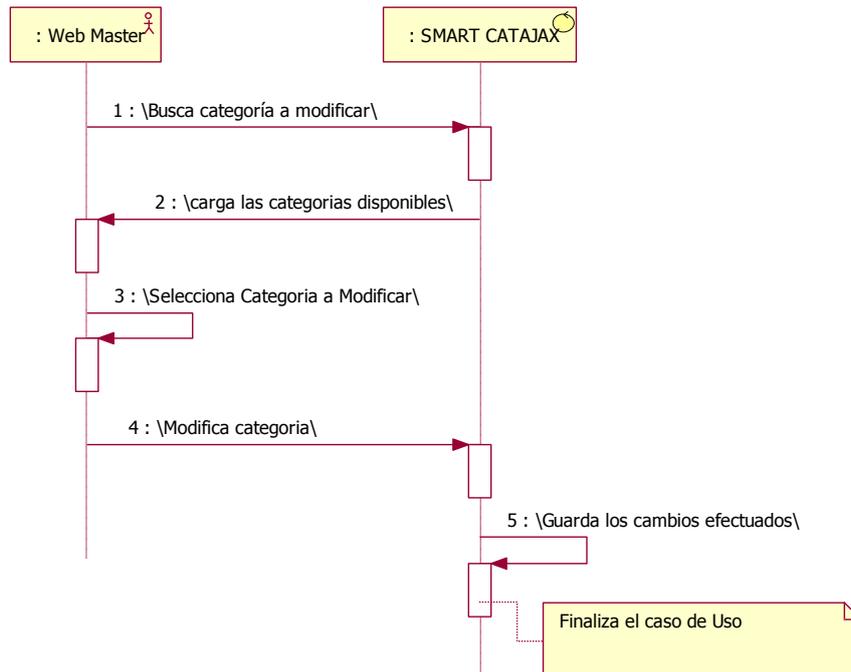
Flujo Básico: Gestionar Categorías



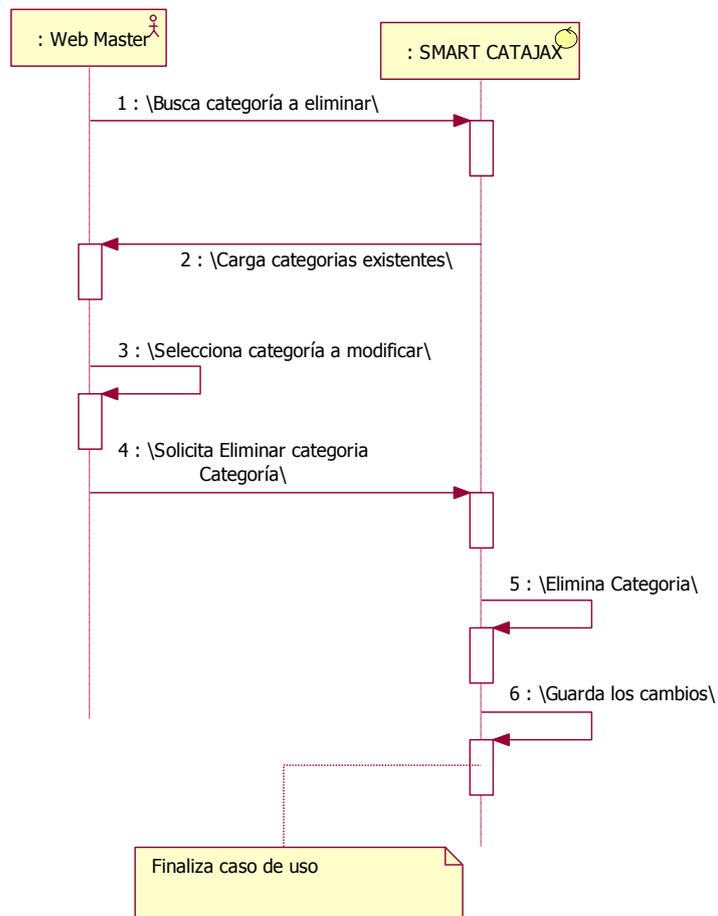
Sub flujo: Ingresar Categorías



Sub flujo: Modificar Categorías

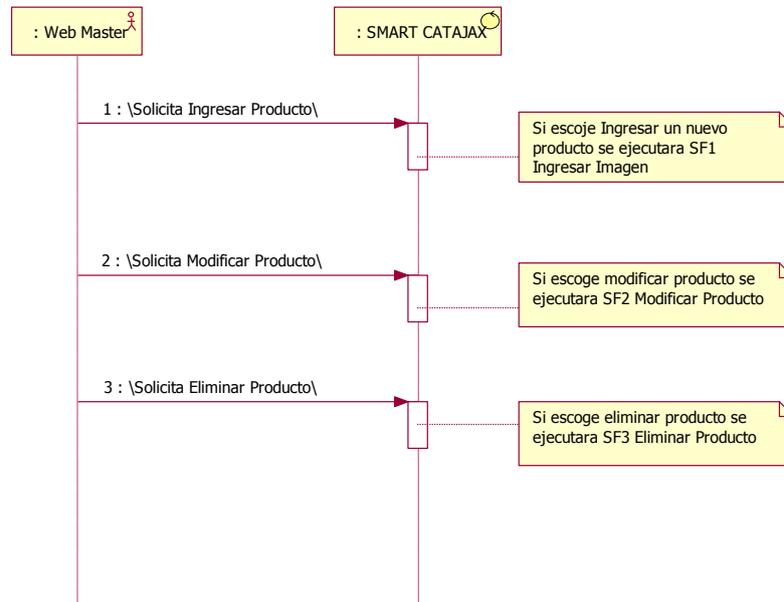


Sub flujo: Eliminar Categorías

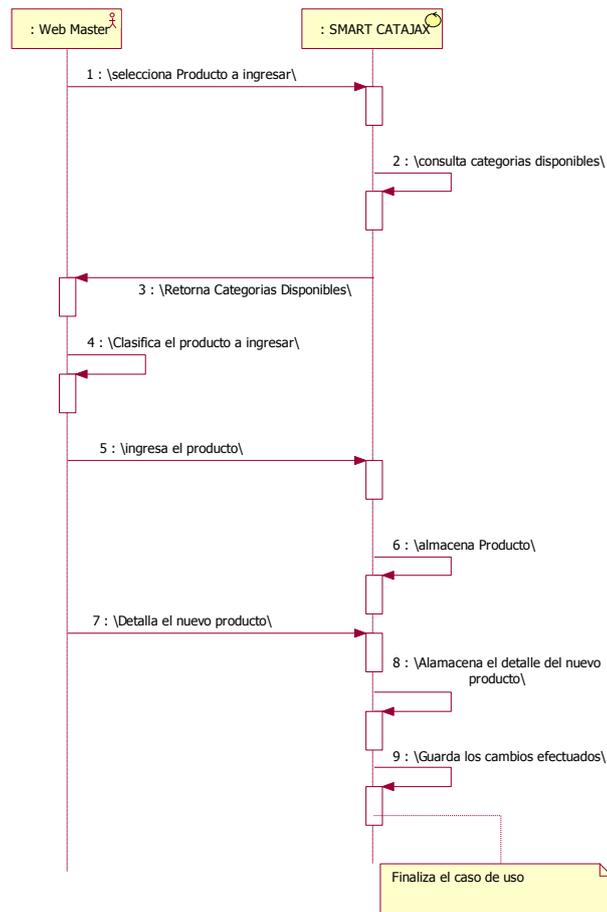


7.3.6 Gestionar Productos

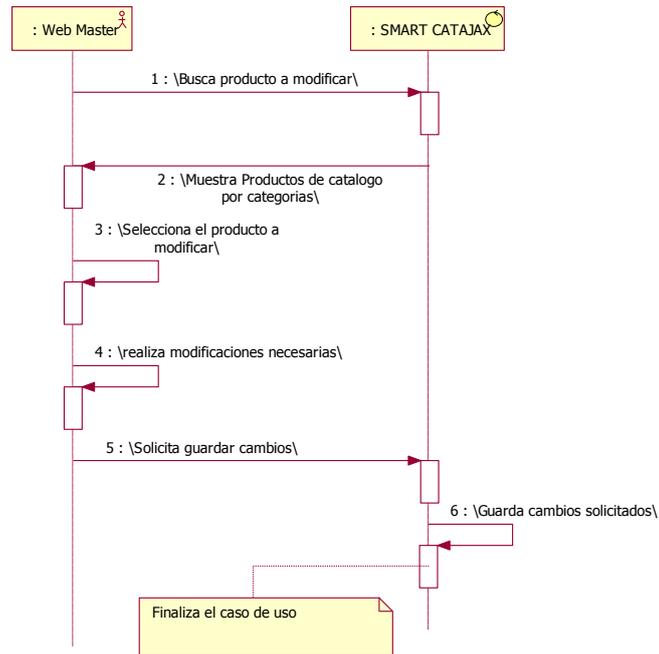
Flujo Básico: Gestionar Productos



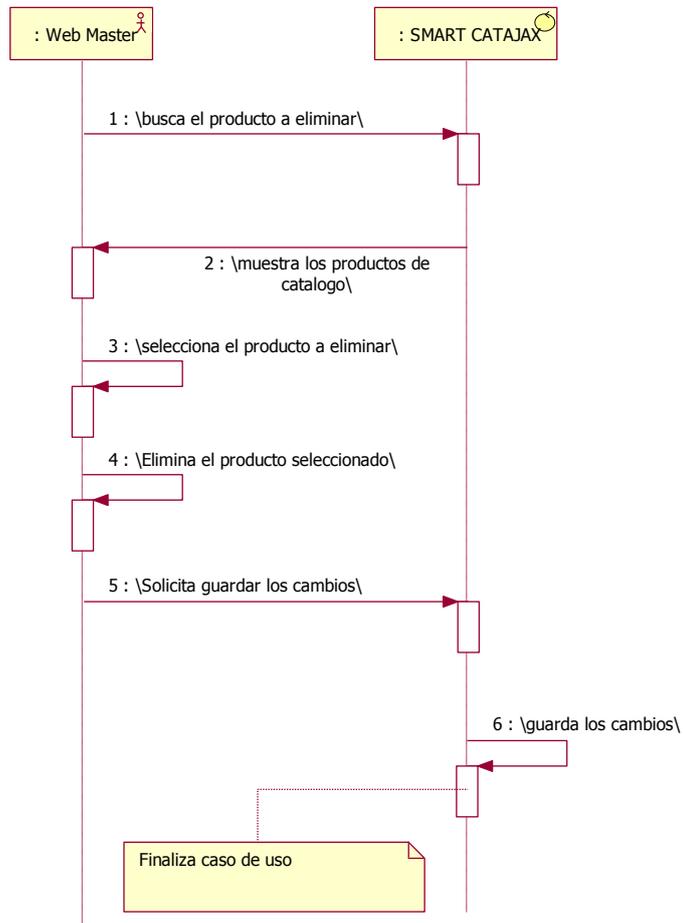
Sub flujo: Ingresar Productos



Sub flujo: Modificar Productos

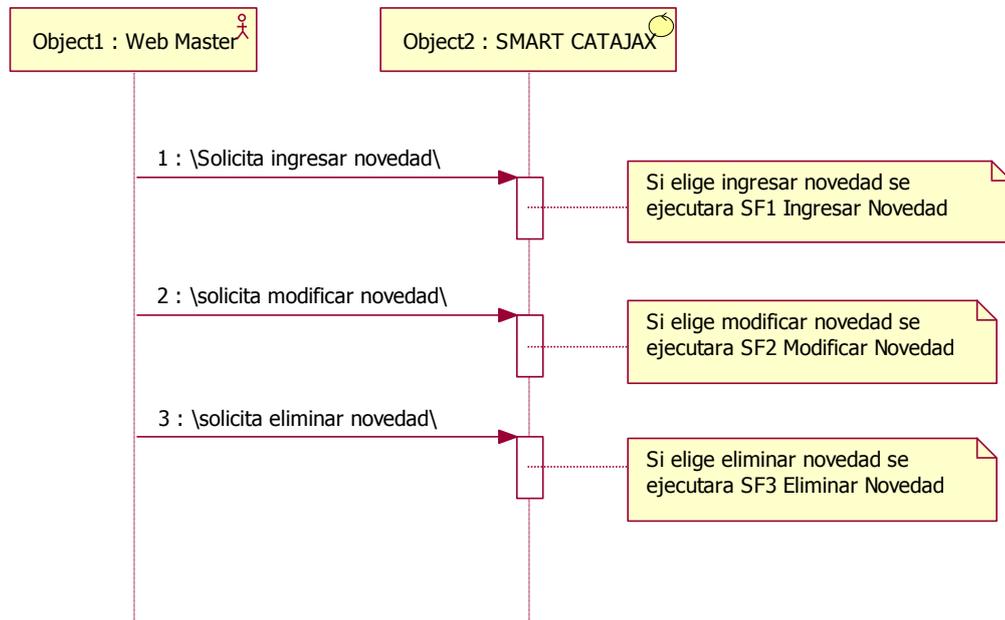


Sub flujo: Eliminar Productos

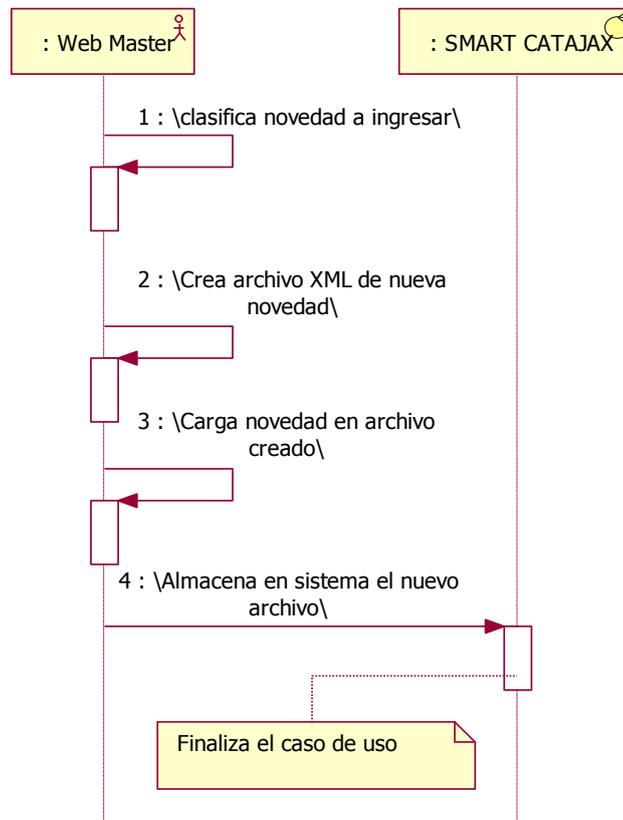


7.3.7 Gestionar Novedades

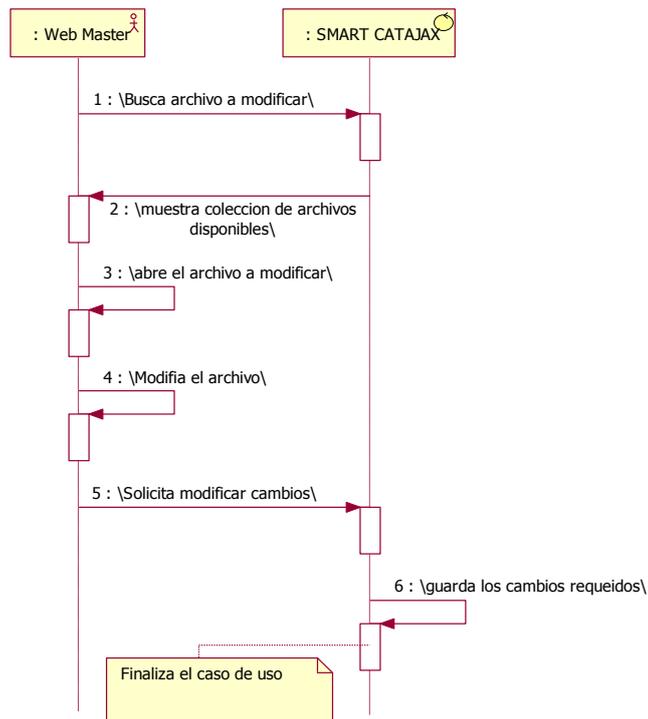
Flujo Básico: Ingresar Novedad



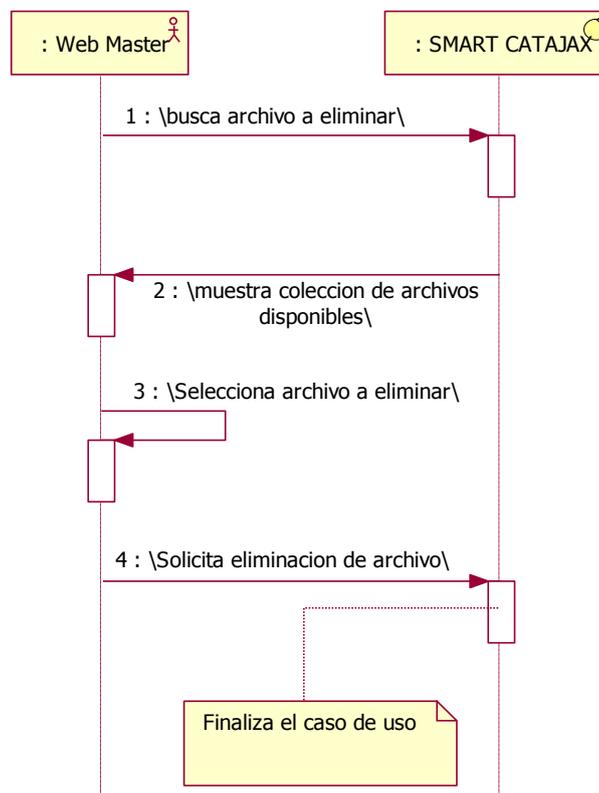
Sub flujo: Insertar Novedad



Sub flujo: Modificar Novedad



Sub flujo: Eliminar Novedad



7.4. Anexo “D”: Diagramas De Colaboración De Análisis

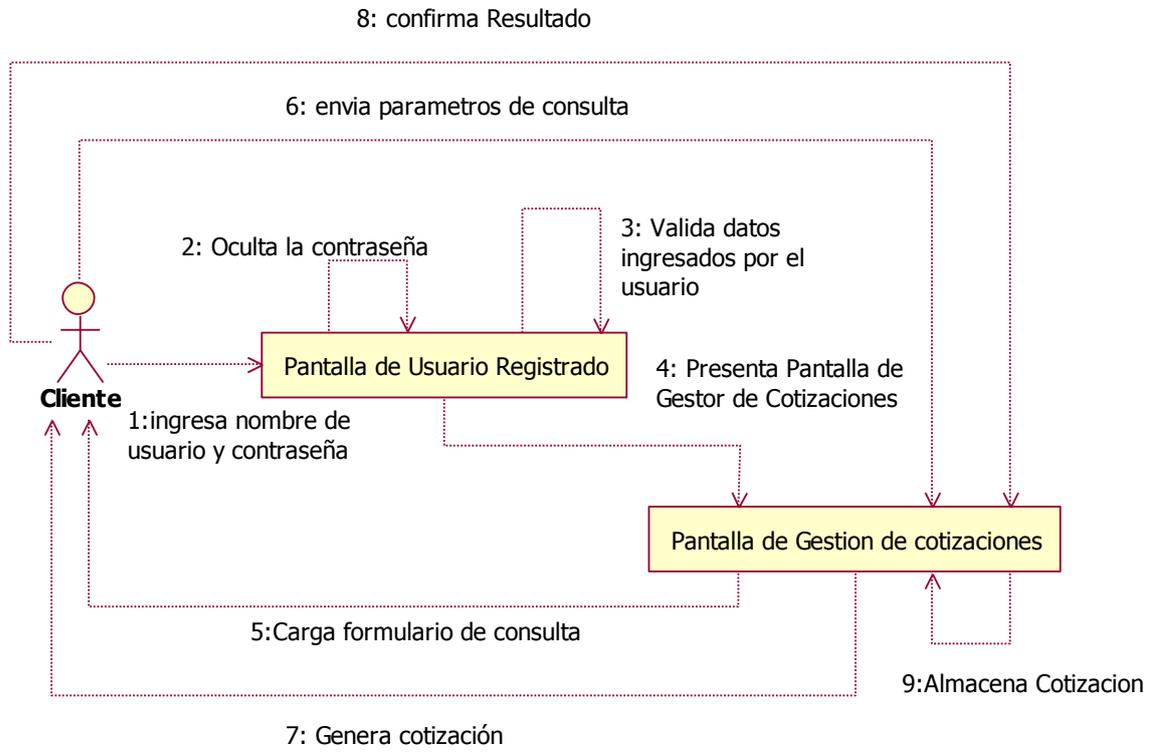
DIAGRAMAS DE COLABORACIÓN DE ANÁLISIS

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

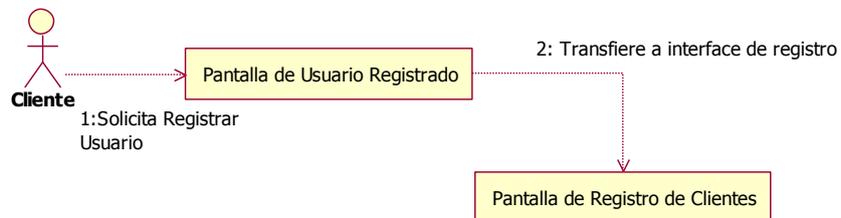
Versión 1.2.0

7.4.1 Gestionar Cotizaciones

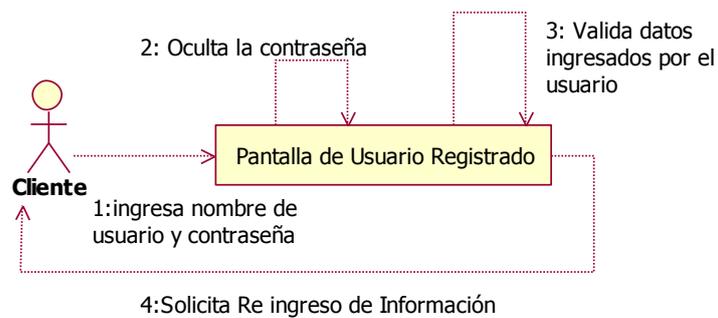
Flujo Básico: Gestionar Cotizaciones



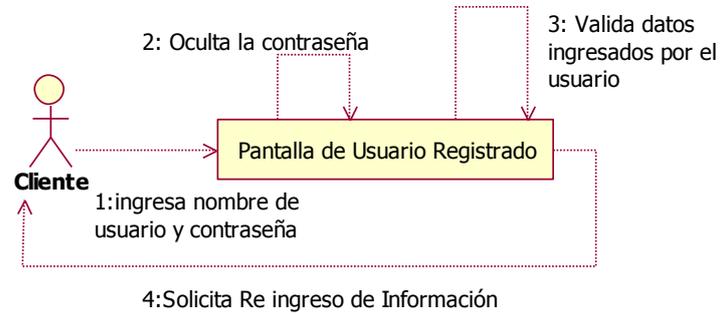
Flujo Alterno FA1: Registrar Usuario



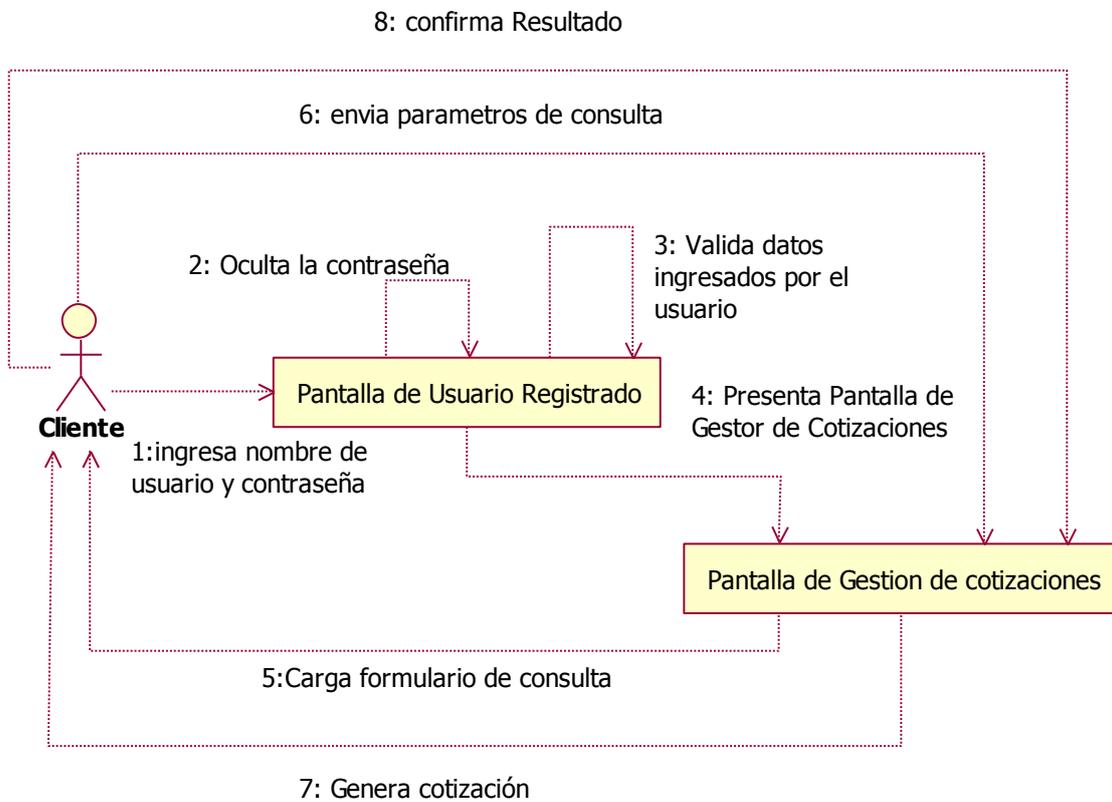
Flujo de Excepción FE1: No ingresa datos el cliente



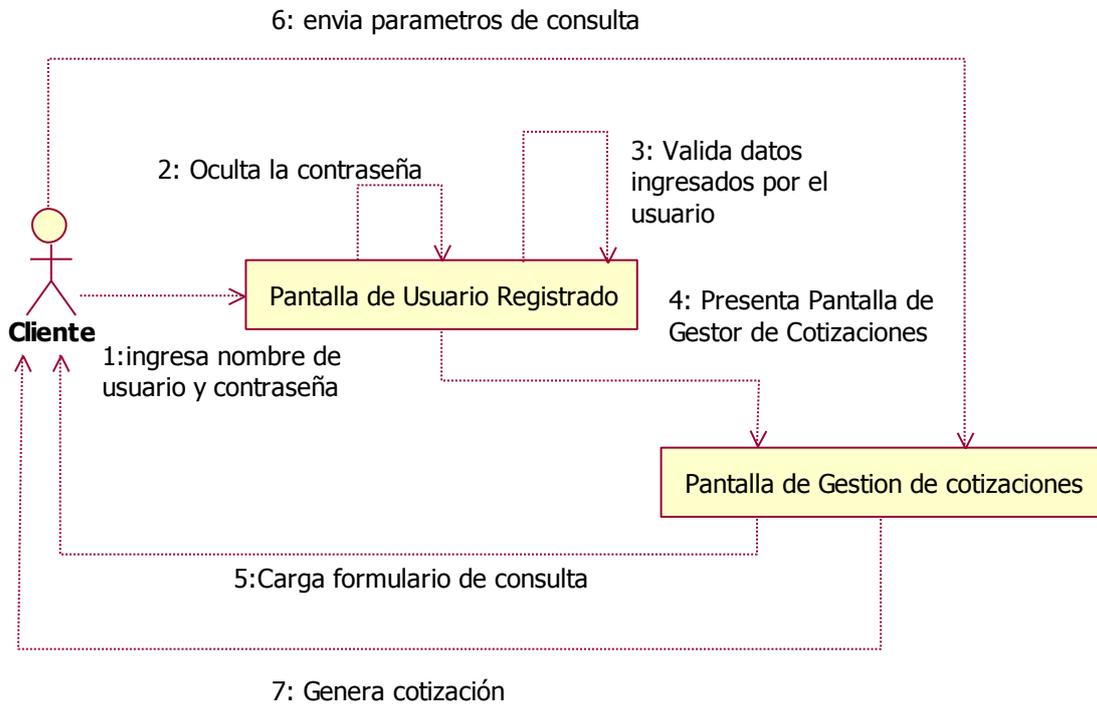
Flujo Alternativo FA2: El cliente ingresa mal sus datos



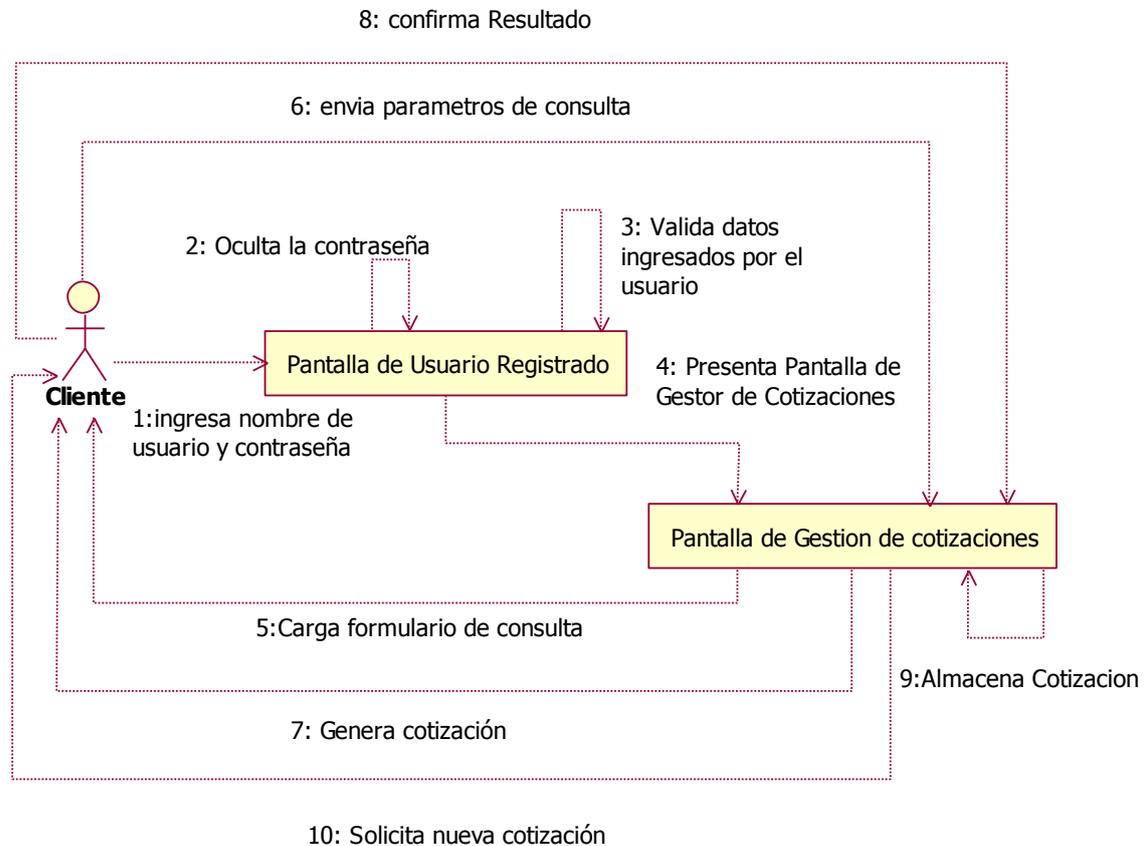
Flujo Alternativo FA3: El cliente confirma su respuesta en desacuerdo



Flujo Excepción FA2: EL usuario sale de la aplicación

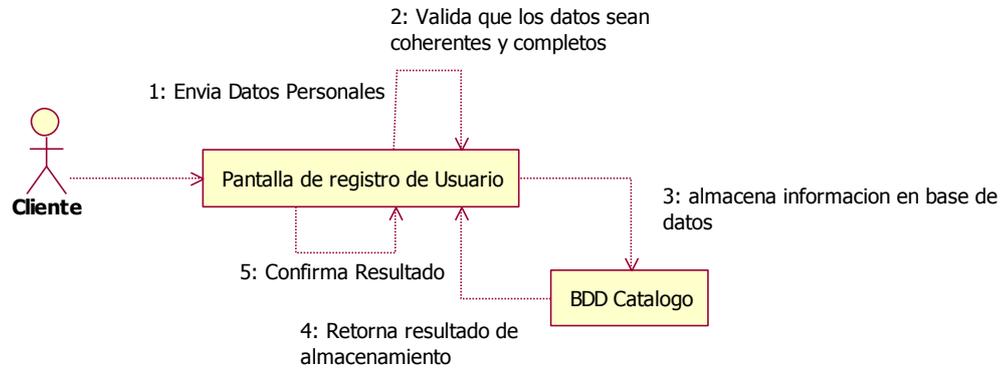


Flujo Alternativo FA4: El usuario escoge continuar con otra cotización

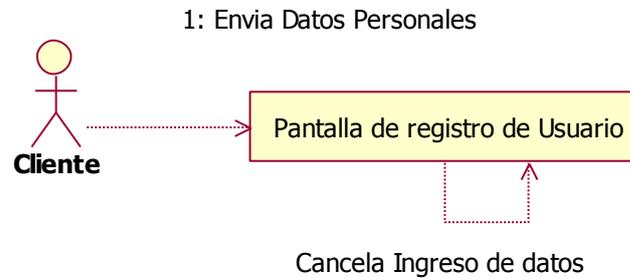


7.4.2 Registrar Usuario

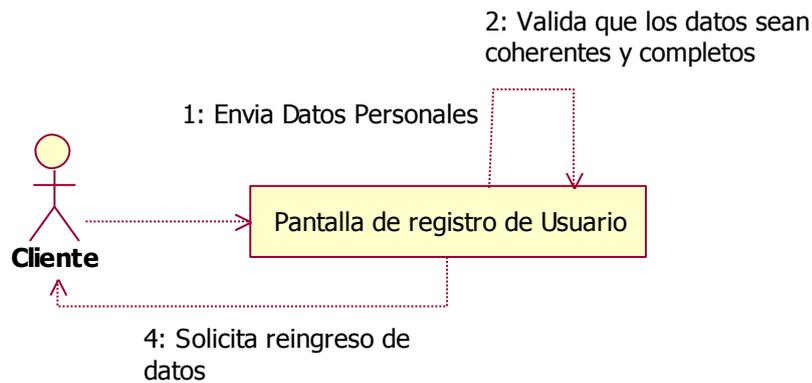
Flujo Básico: Registrar Usuario



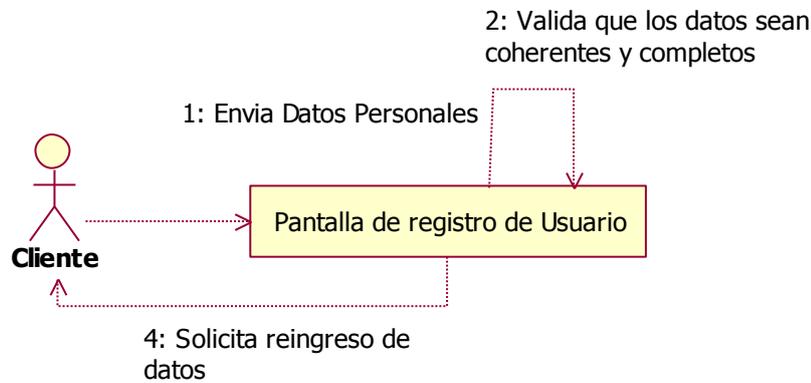
Flujo de Excepción FE1: El usuario Cancela el ingreso



Flujo de Excepción FE2: El usuario Ingresa mal sus datos

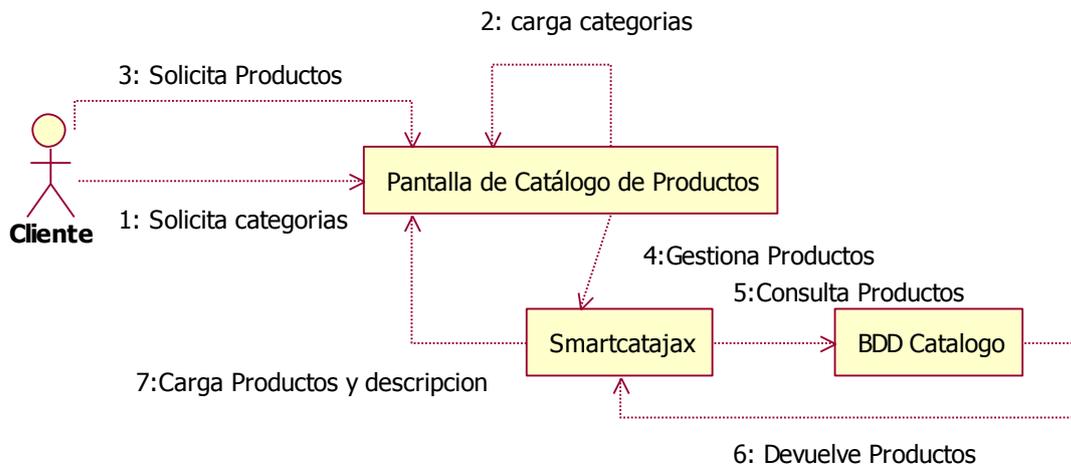


Flujo Alternativo FA1: El usuario no ingreso a información completa o correcta para la creación de un nuevo Usuario

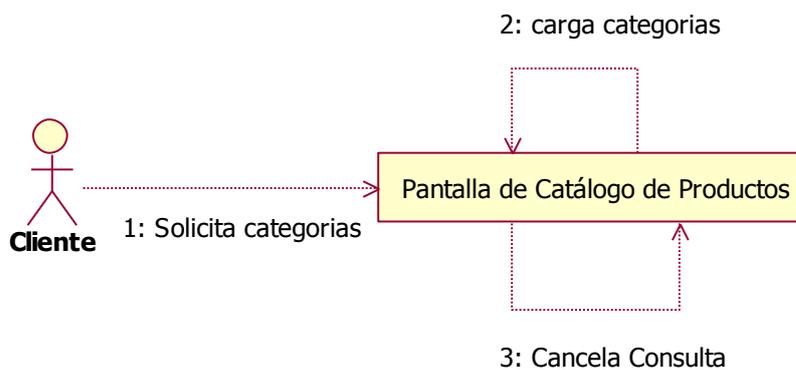


7.4.3 Consultar Catálogo

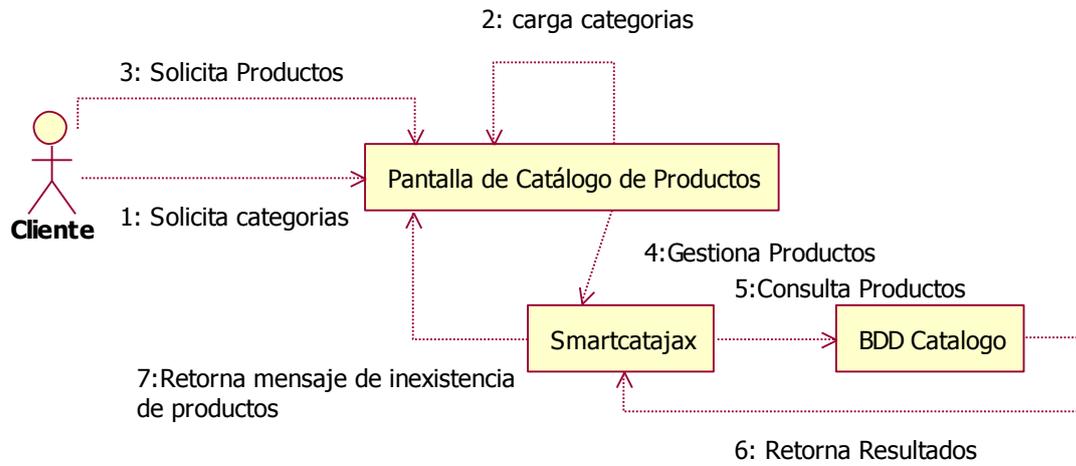
Flujo Básico: Consultar Catálogo



Flujo de excepción FE: Usuario Cancela Consulta

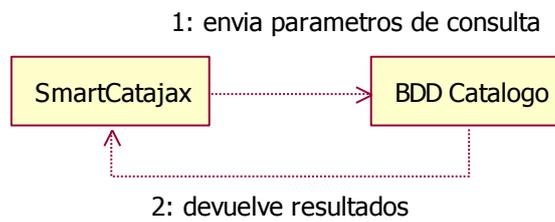


Flujo de excepción FA: No Existen Productos



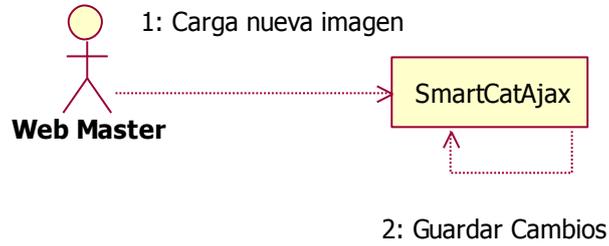
7.4.4 Validar Usuario

Flujo Básico: Validar Usuario

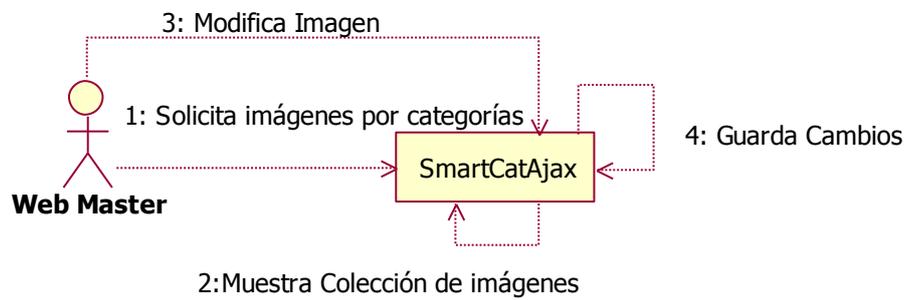


7.4.5 Gestionar Imágenes

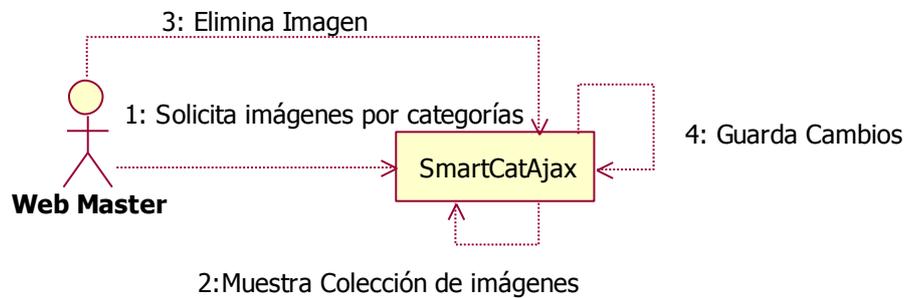
Sub flujo: Ingresar Imagen



Sub flujo: Modificar Imagen

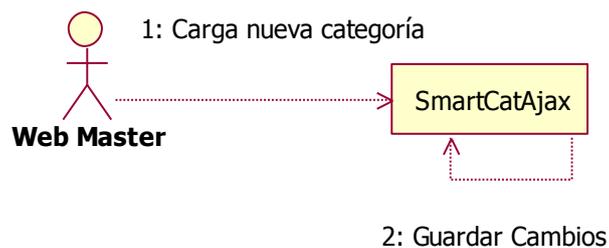


Sub flujo: Eliminar Imagen

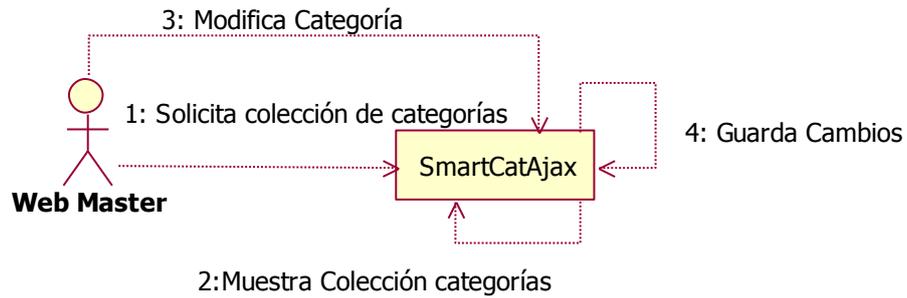


7.4.6 Gestionar Categorías

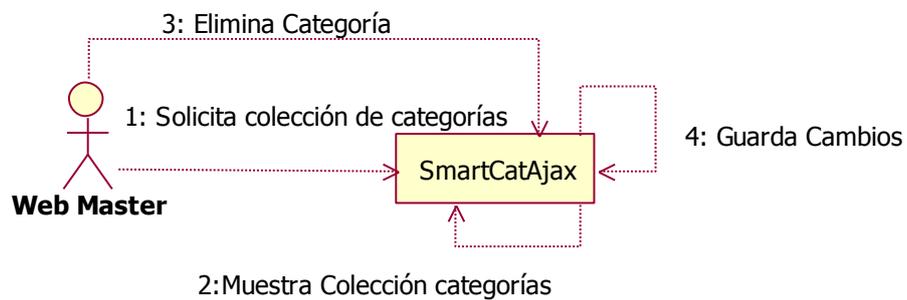
Sub flujo: Ingresar Categoría



Sub flujo: Modificar Categoría

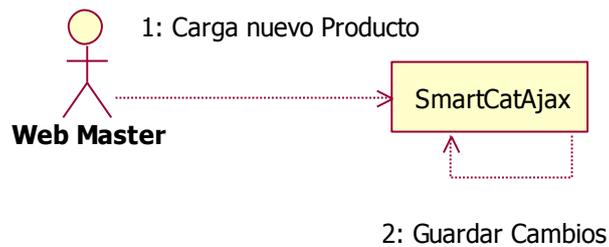


Sub flujo: Eliminar Categoría

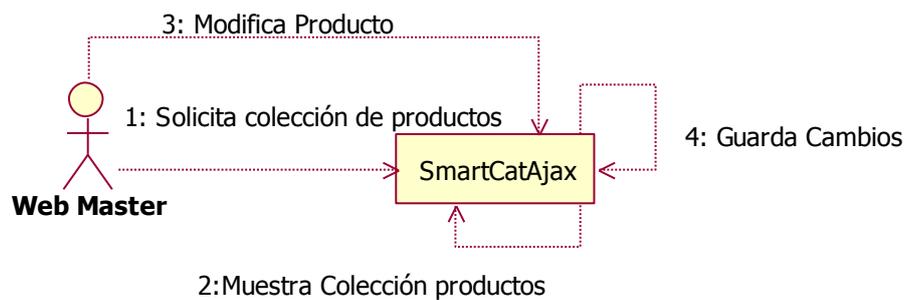


7.4.7 Gestionar Productos

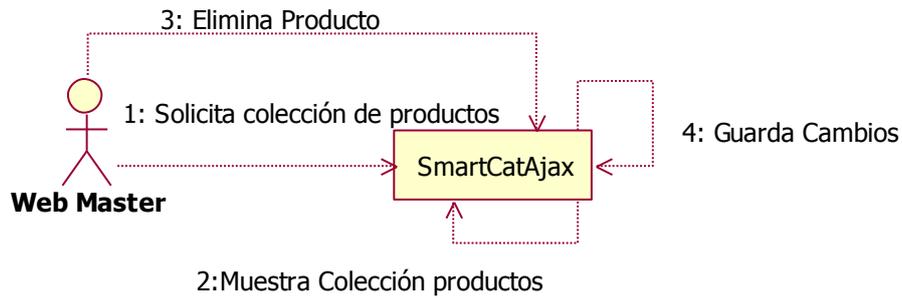
Sub flujo: Ingresar Producto



Sub flujo: Modificar Producto

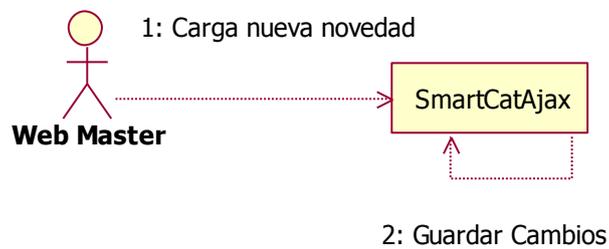


Sub flujo: Eliminar Producto

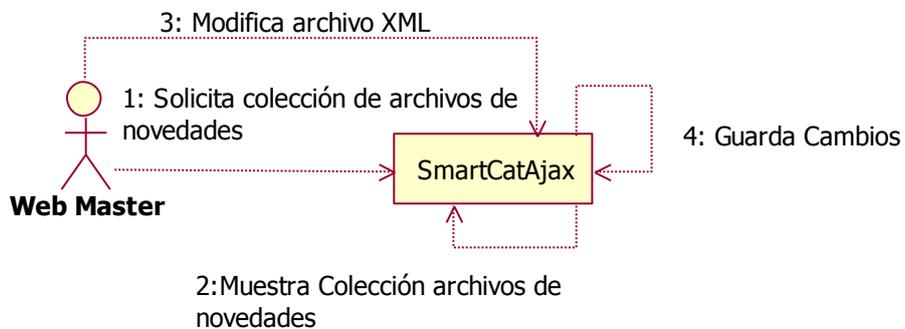


7.4.8 Gestionar Novedades

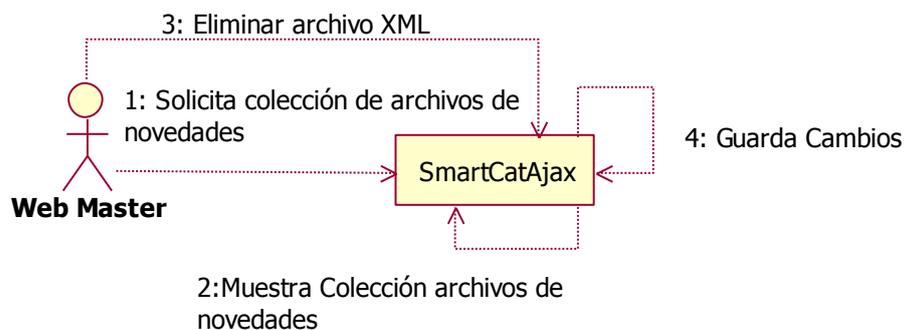
Sub flujo: Ingresar Novedad



Sub flujo: Modificar Novedad



Sub flujo: Eliminar Novedad



7.5. Anexo “E”: Diagramas De Colaboración De Diseño

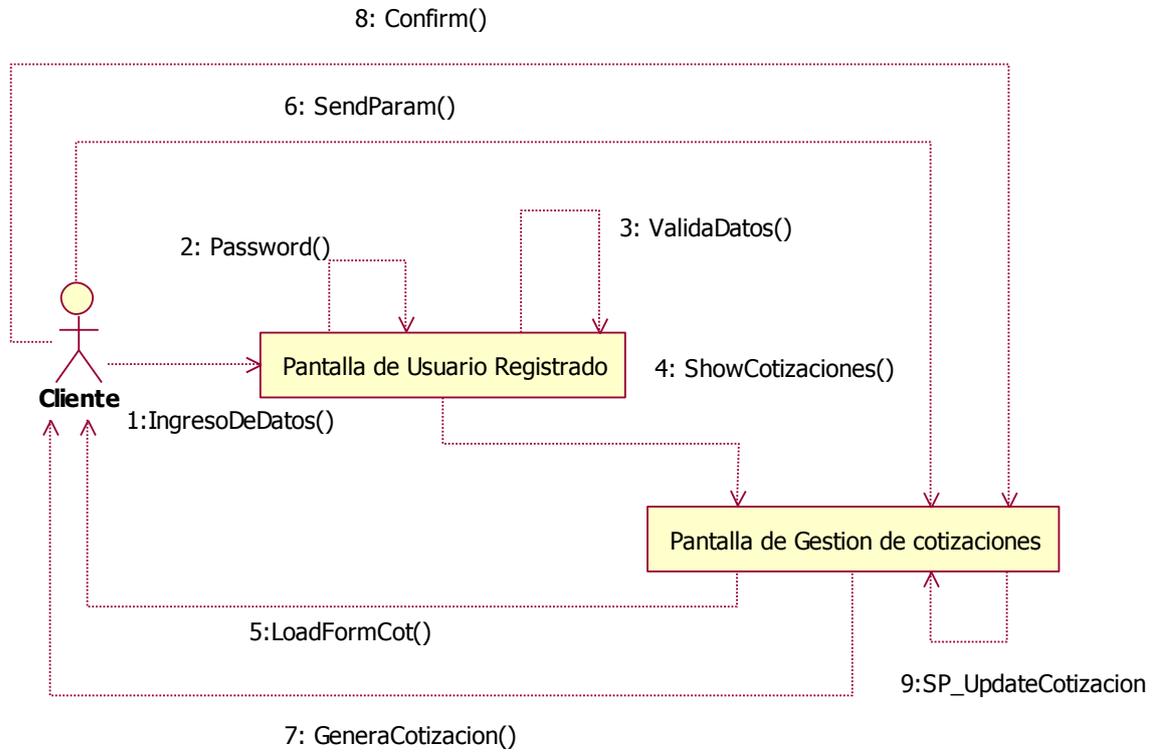
DIAGRAMAS DE COLABORACIÓN DE DISEÑO

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

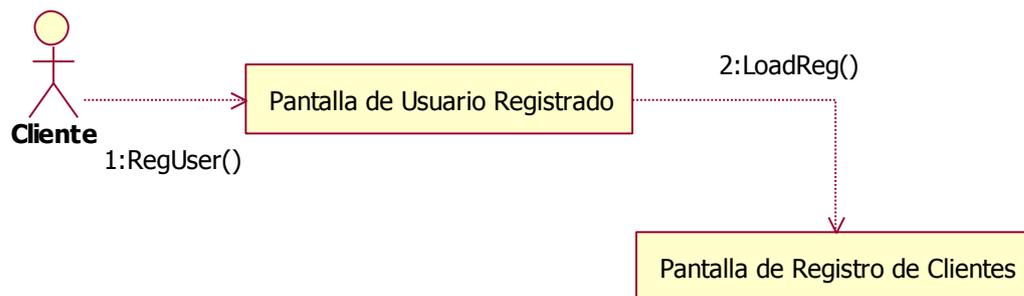
Versión 1.2.0

7.5.1 Gestionar Cotizaciones

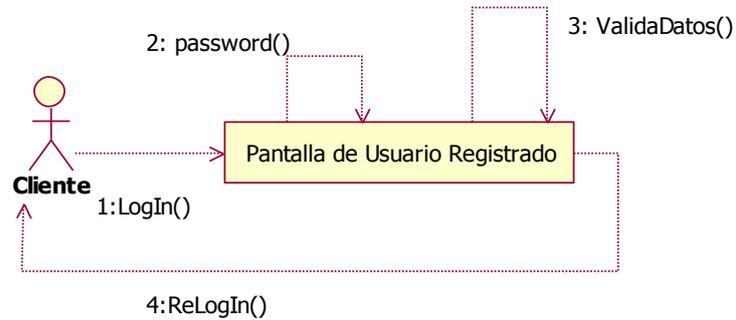
Flujo Básico: Gestionar Cotizaciones



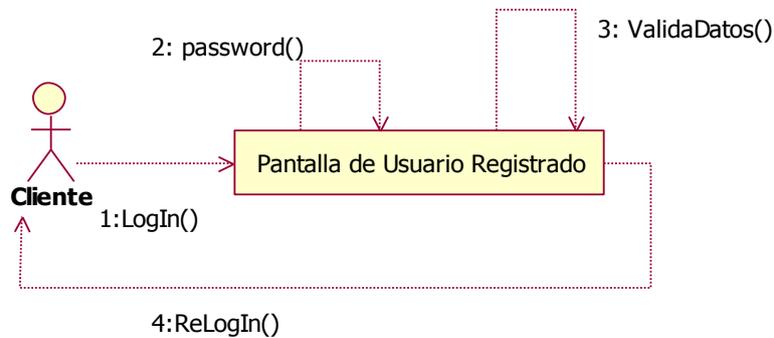
Flujo Alternativo FA1: Registrar Cliente



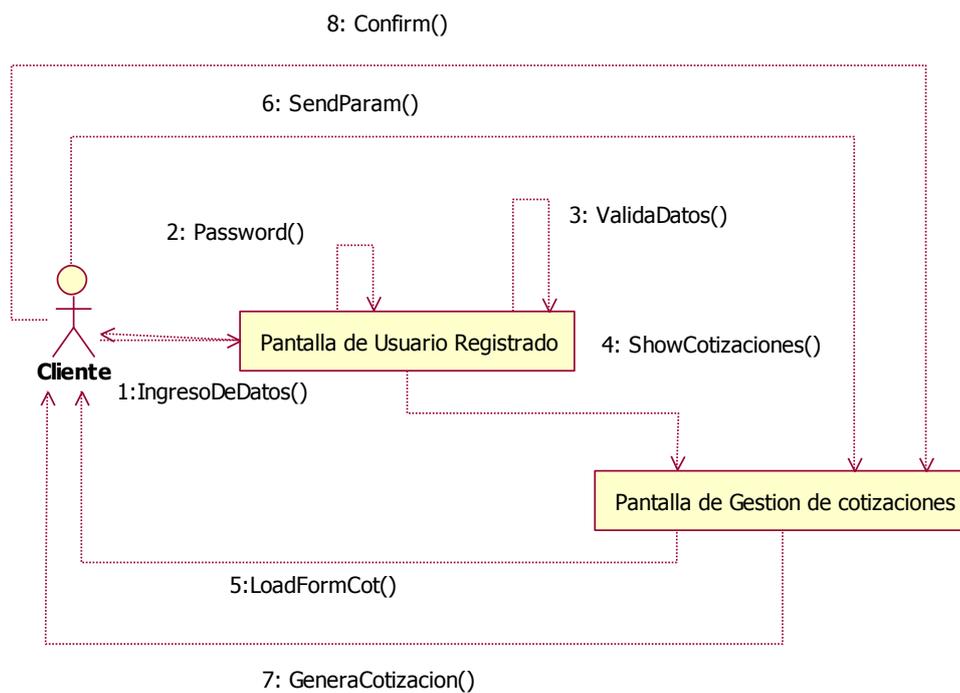
Flujo de Excepción FE1: No ingresa datos el cliente



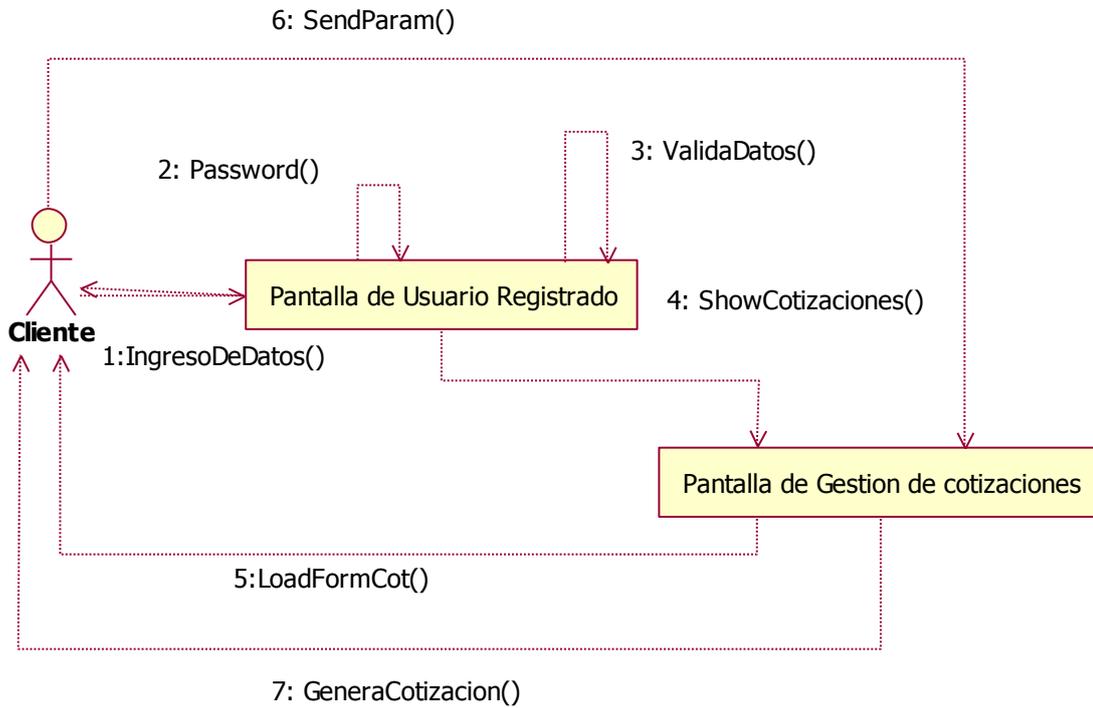
Flujo Alternativo FA2: El cliente ingresa mal sus datos



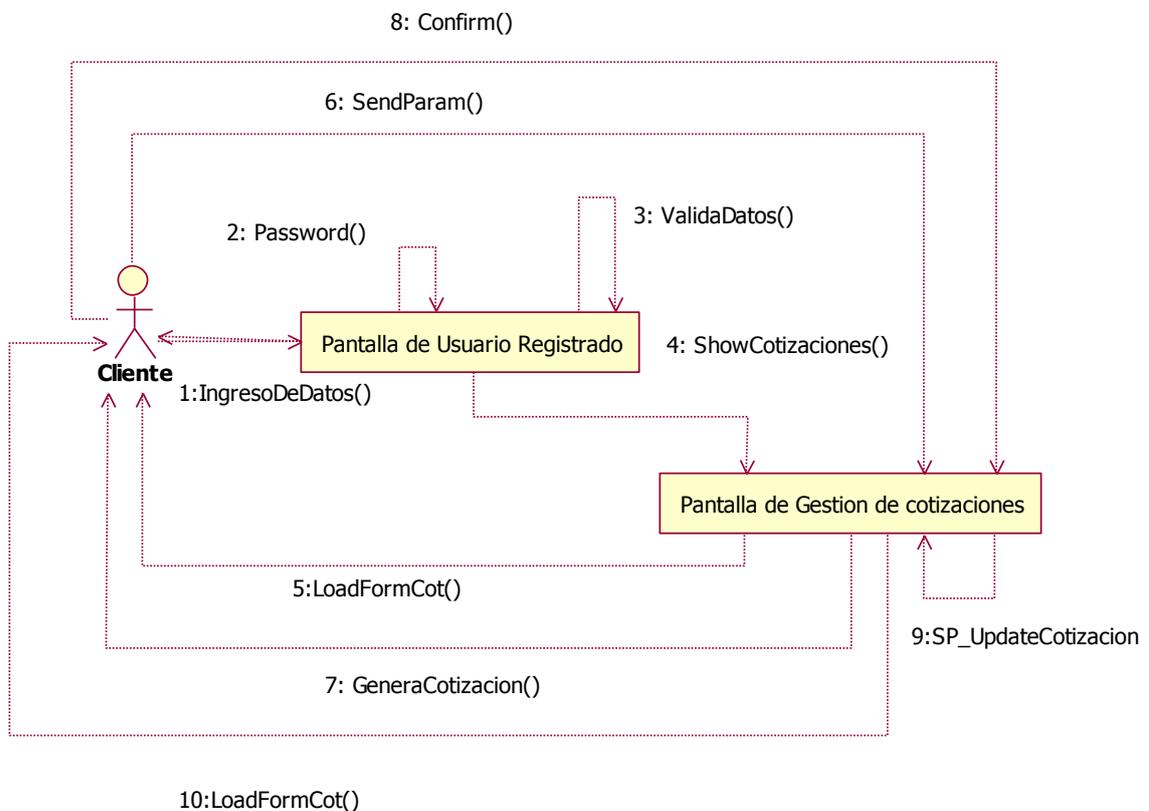
Flujo Alternativo FA3: El cliente confirma su respuesta en desacuerdo



Flujo Excepción FA2: EL usuario sale de la aplicación

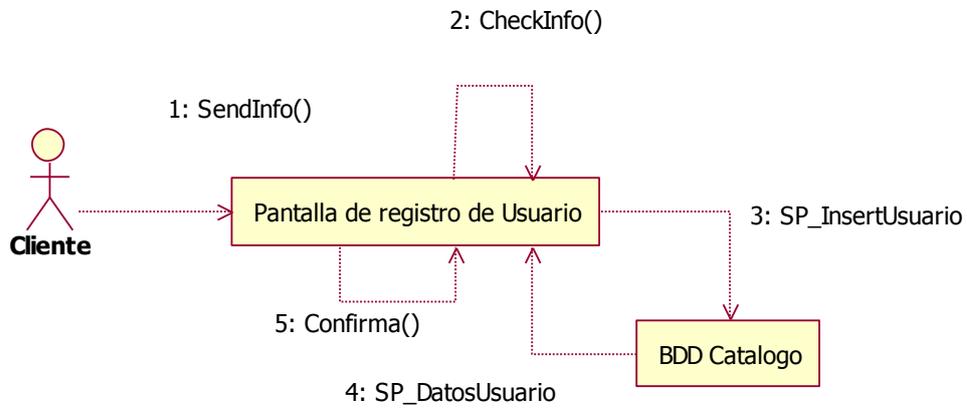


Flujo Alternativo FA4: El usuario escoge continuar con otra cotización

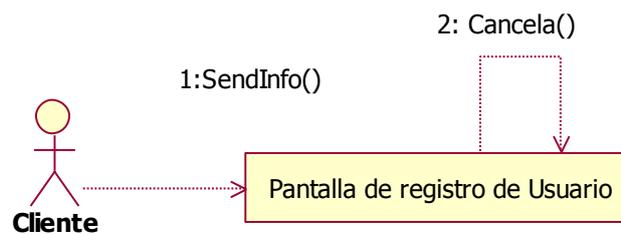


7.5.2 Registrar Usuario

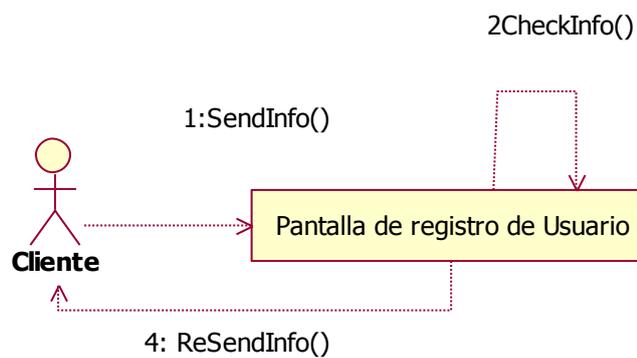
Flujo Básico: Registrar Usuario



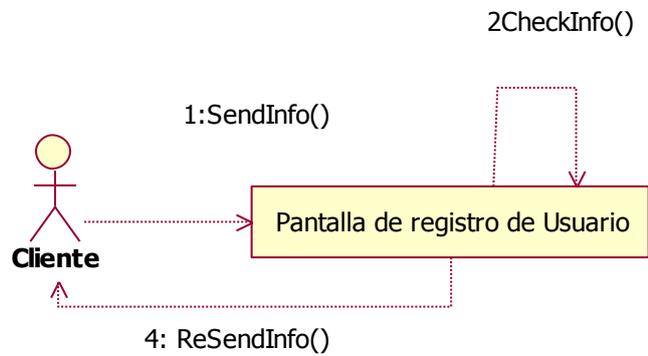
Flujo de Excepción FE1: El usuario Cancela el ingreso



Flujo de Excepción FE2: El usuario Ingresa mal sus datos

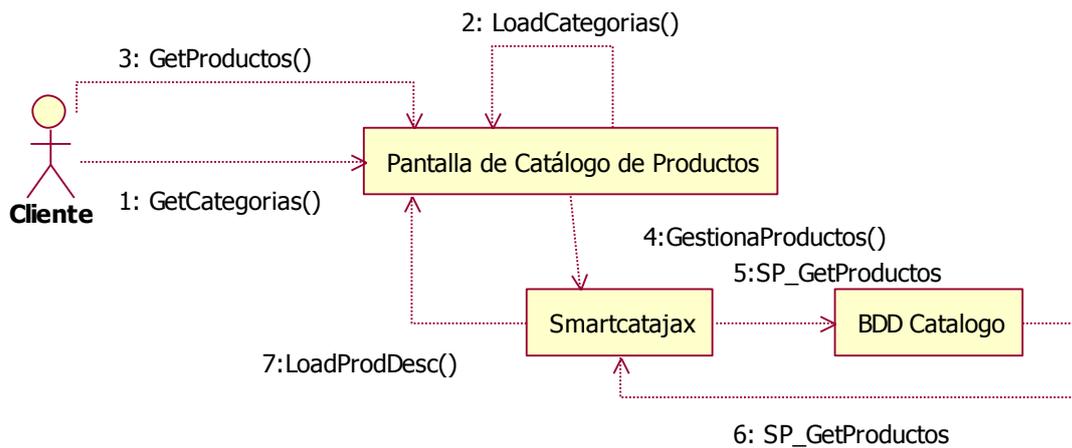


Flujo Alterno FA1: El usuario no ingreso a información completa o correcta para la creación de un nuevo Usuario

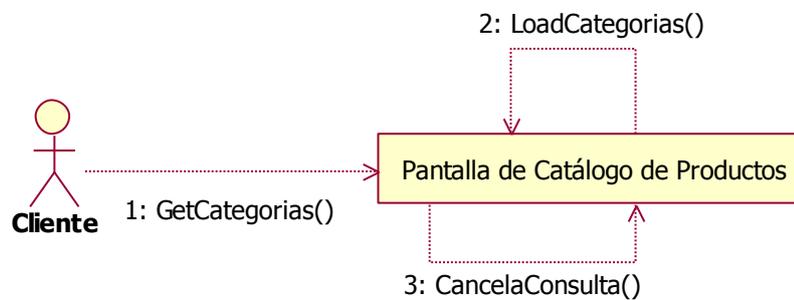


7.5.3 Consultar Catálogo

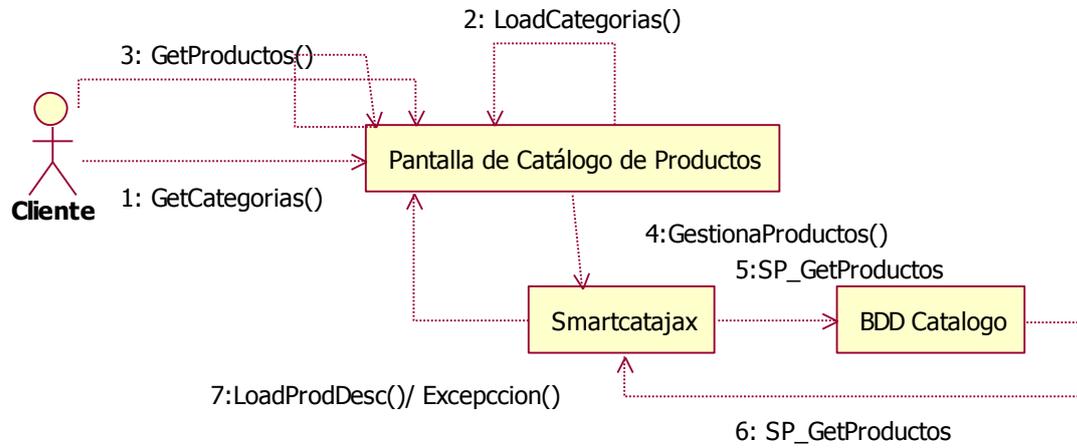
Flujo Básico: Consultar Catálogo



Flujo de excepción FE: Usuario Cancela Consulta

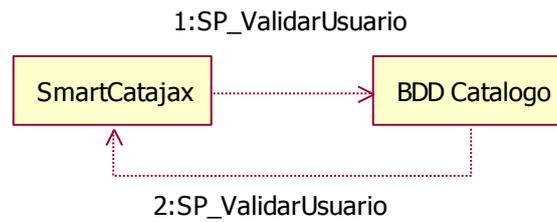


Flujo de excepción FA: No Existen Productos



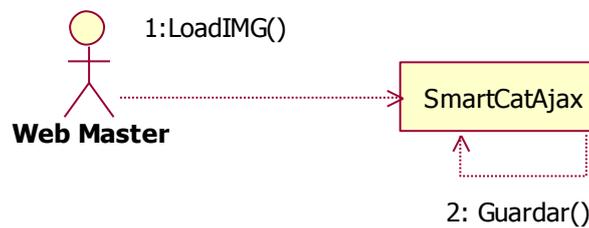
7.5.4 Validar Usuario

Flujo Básico: Validar Usuario

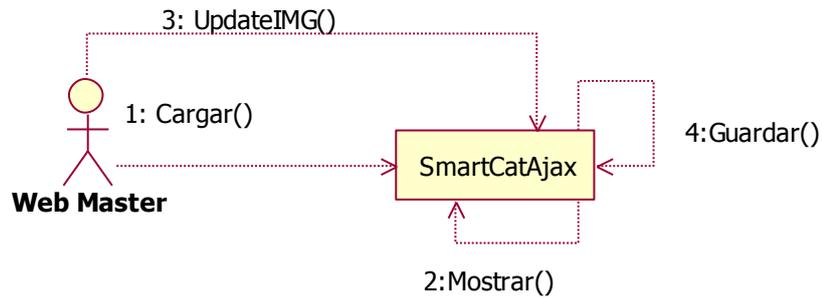


7.5.5 Gestionar Imágenes

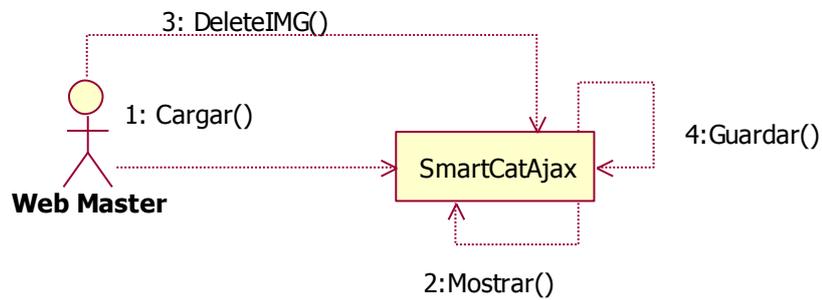
Sub flujo: Ingresar Imagen



Sub flujo: Modificar Imagen

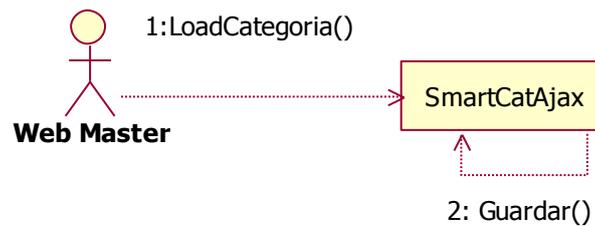


Sub flujo: Eliminar Imagen

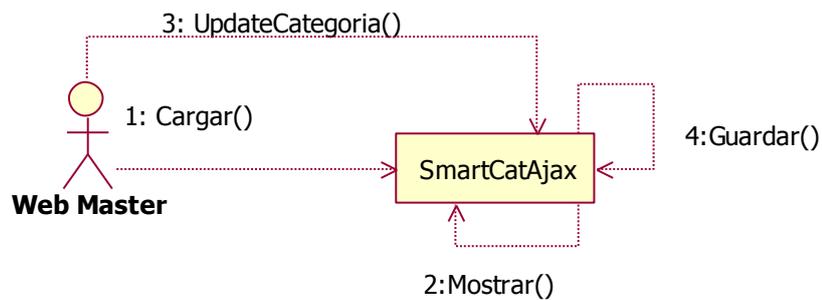


7.5.6 Gestionar Categorías

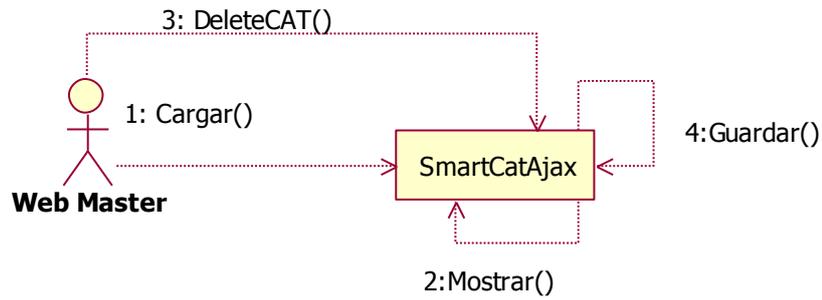
Sub flujo: Ingresar Categorías



Sub flujo: Modificar Categorías

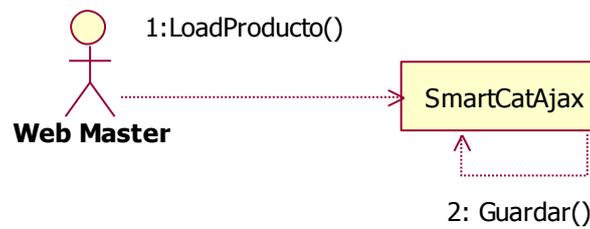


Sub flujo: Eliminar Categorías

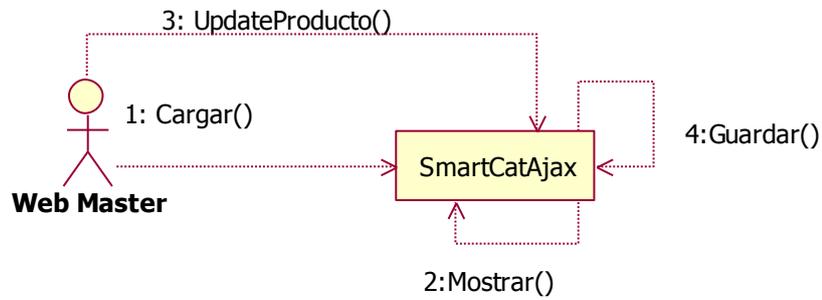


7.5.7 Gestionar Productos

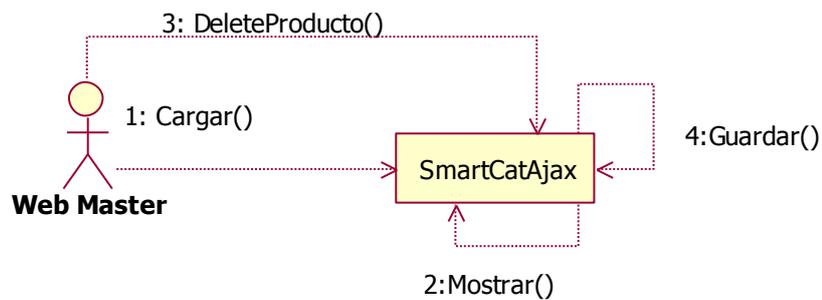
Sub flujo: Ingresar Productos



Sub flujo: Modificar Productos

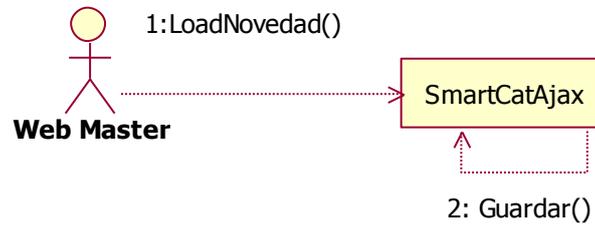


Sub flujo: Eliminar Productos

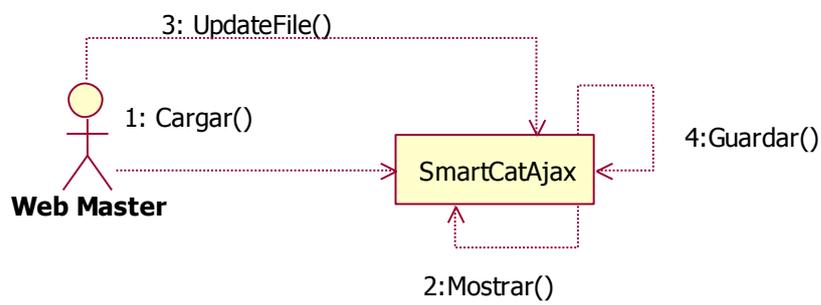


7.5.8 Gestionar Novedades

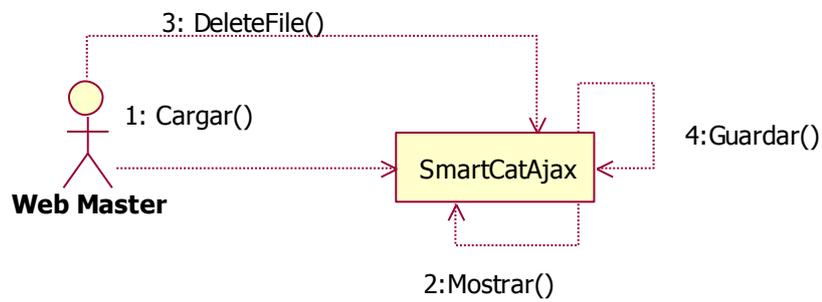
Sub flujo: Ingresar Novedades



Sub flujo: Modificar Novedades



Sub flujo: Eliminar Novedades

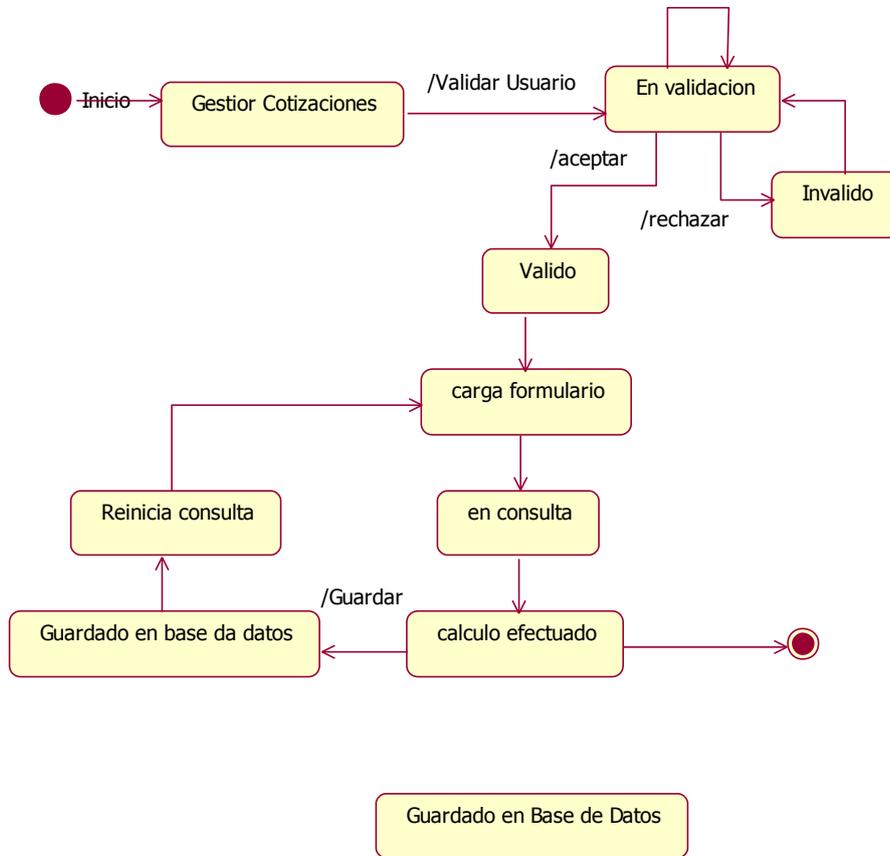


7.6. Anexo “F”: Diagramas De Estado

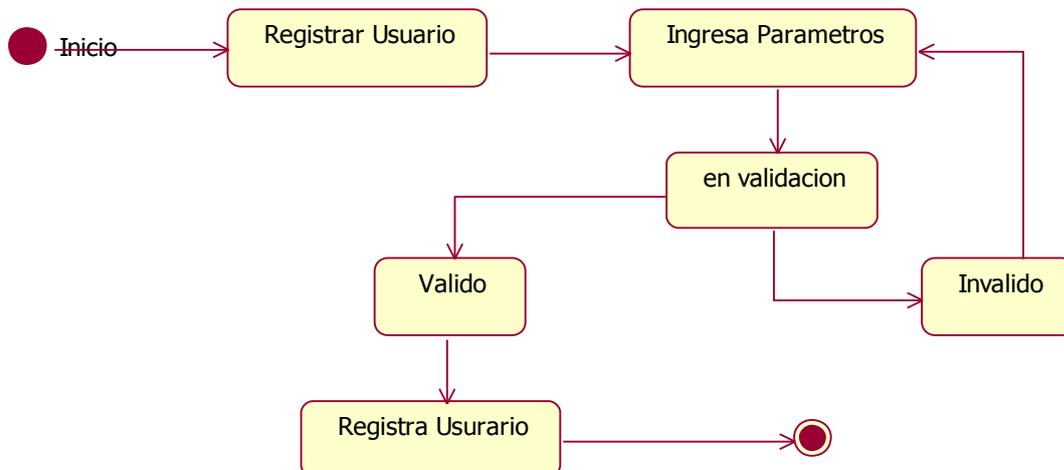
DIAGRAMAS DE ESTADO

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

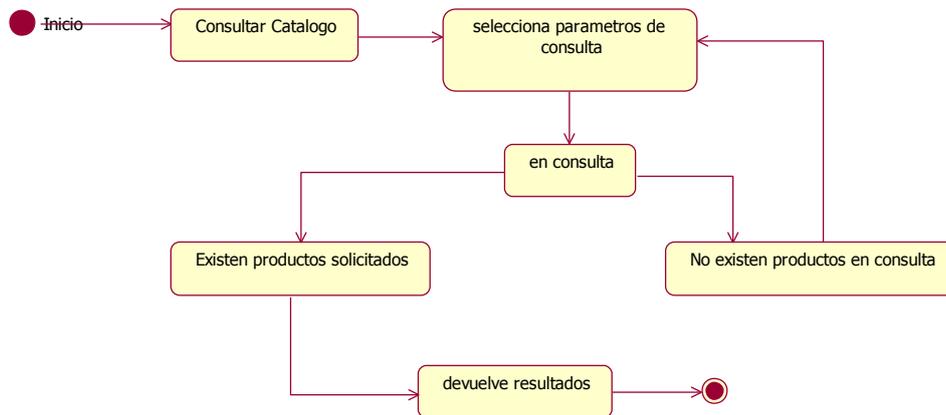
7.6.1 Gestionar Cotizaciones



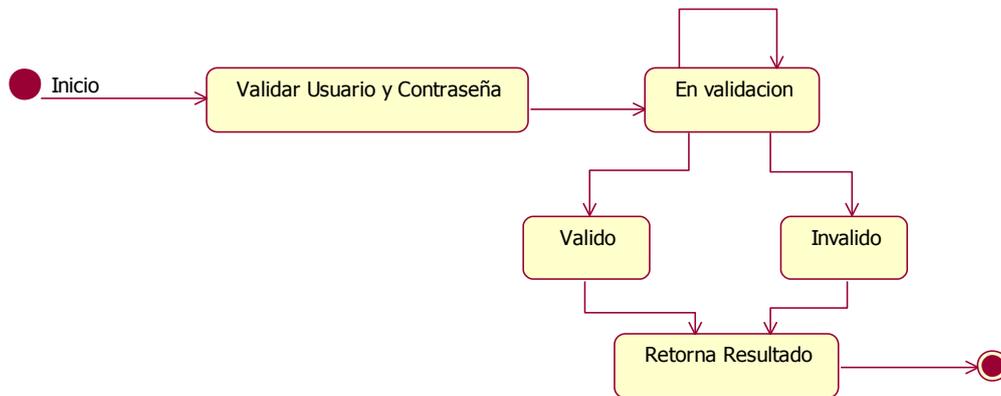
7.6.2 Registrar Usuario



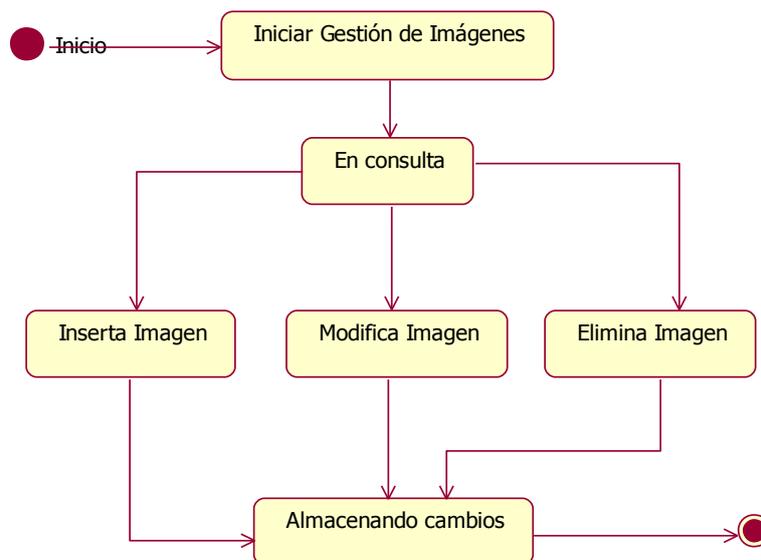
7.6.3 Consultar Catálogo



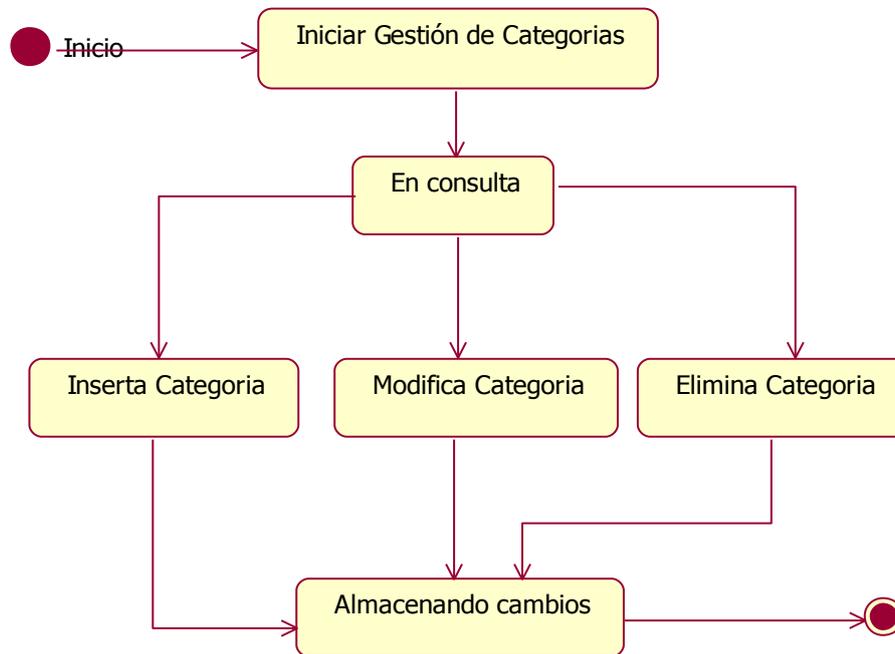
7.6.4 Validar Usuario



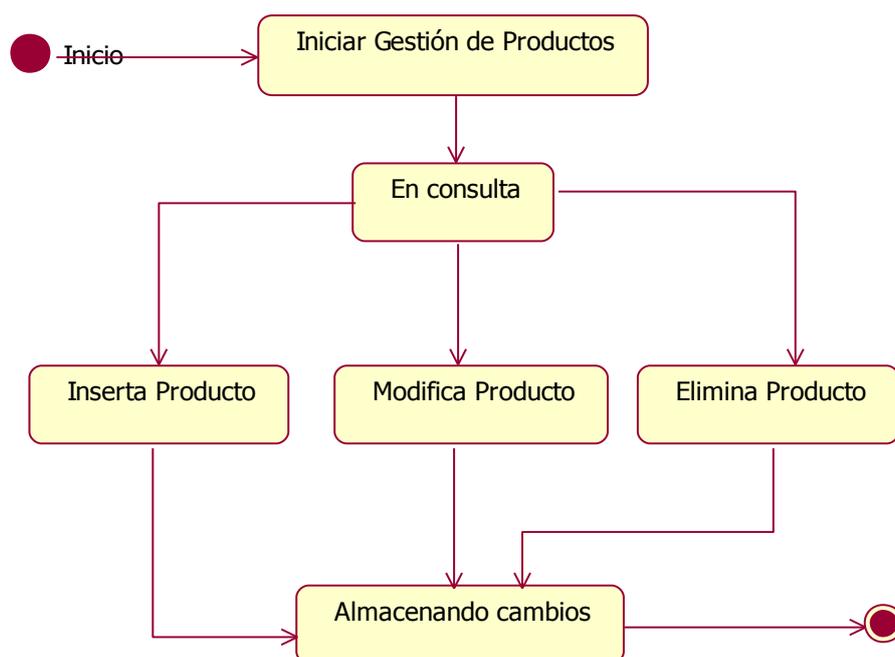
7.6.5 Gestionar Imágenes



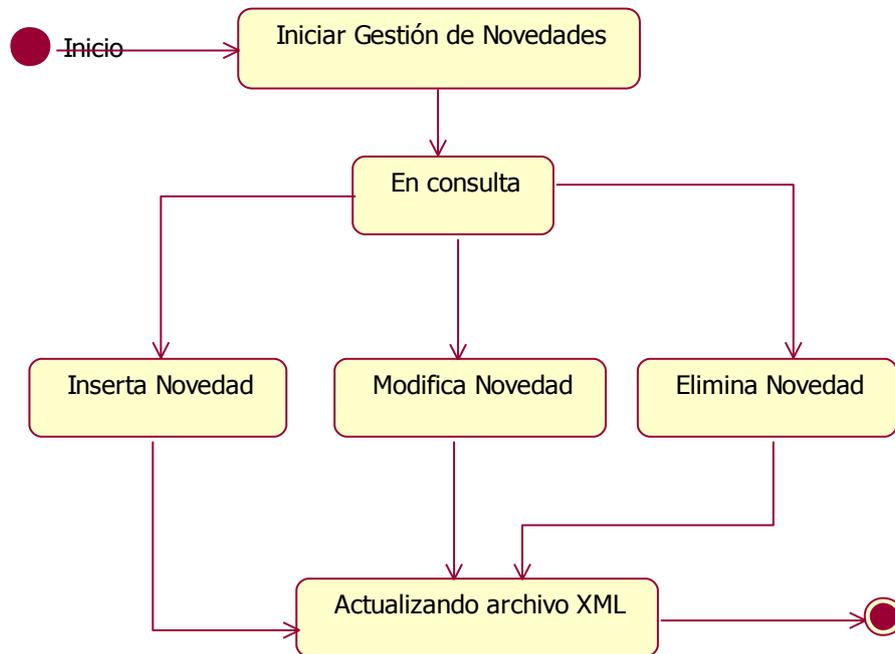
7.6.6 Gestionar Categorías



7.6.7 Gestionar Productos



7.6.8 Gestionar Novedades



7.7. Anexo “G”: Implementación De Código

IMPLEMENTACIÓN DE CÓDIGO

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

Versión 1.2.0

7.8.1 Implementación De Código

Antes de la programación de los servicios que cumplirán con las funciones requeridas para el Sistema de Catálogo en Línea; se debe escribir los flujos de donde saldrán los servicios a implementar.

Las operaciones y funciones se implementan según el siguiente detalle:

- Programación Multicapas en ASP.NET con conexión a MS SQL Express
- Interface de usuario multimedia con tecnología Silverlight mediante XAML y soporte de JavaScript y ASP.NET.
- Interface de usuario multimedia con tecnología Flash, ActionScript 2.0 y XML y HTML

Un ejemplo de ello se da cuando un Usuario solicita ingresar al sistema de cotizaciones; se invoca al módulo de “Gestión de cotizaciones”; el mismo que a su vez invoca al módulo de Validación de Usuario el cual se encarga de validar la existencia de dicho usuario, ejecutando las siguientes operaciones:

1. El sistema envía los datos ingresados por el usuario
2. El sistema recibe los datos de la aplicación.
3. El sistema valida los datos ingresados mediante la ejecución de una consulta a la base de datos del sistema.
4. El sistema retorna el resultado de la validación.

El flujo básico detallado; es convertido en código mediante la siguiente secuencia de fases.

7.8.2 FASE 1 – ASP.NET, Flash

Código ASP.NET

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>homednp08</title>
<script language="javascript">AC_FL_RunContent = 0;</script>
<script src="AC_RunActiveContent.js" language="javascript"></script>
</head>
<body bgcolor="#131415">
<!--url's used in the movie-->
<!--text used in the movie-->
<!--
<p align="justify"></p>
-->
<!-- saved from url=(0013)about:internet -->
<script language="javascript">
    if (AC_FL_RunContent == 0) {
        alert("This page requires AC_RunActiveContent.js.");
    } else {
        AC_FL_RunContent(
            'codebase',
'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0',
            'width', '760',
            'height', '480',
            'src', 'homednp08',
            'quality', 'high',
            'pluginspage', 'http://www.macromedia.com/go/getflashplayer',
            'align', 'middle',
            'play', 'true',
            'loop', 'true',
            'scale', 'showall',
```

```

        'wmode', 'window',
        'devicefont', 'false',
        'id', 'homednp08',
        'bgcolor', '#131415',
        'name', 'homednp08',
        'menu', 'true',
        'allowFullScreen', 'false',
        'allowScriptAccess', 'sameDomain',
        'movie', 'homednp08',
        'salign', ''
    ); //end AC code
}
</script>
<noscript>
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0
,0" width="760" height="480" id="homednp08" align="middle">
    <param name="allowScriptAccess" value="sameDomain" />
    <param name="allowFullScreen" value="false" />
    <param name="movie" value="homednp08.swf" /><param name="quality" value="high" />
<param name="bgcolor" value="#131415" />    <embed src="homednp08.swf" quality="high"
bgcolor="#131415" width="760" height="480" name="homednp08" align="middle"
allowScriptAccess="sameDomain" allowFullScreen="false" type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" />
    </object>
</noscript>
</body>
</html>

```

Código Action Script 2.0

```

on (press)
{
    getUrl("http://localhost:52507/ICR.aspx", "_blank");
    menu=5;
    gotoAndPlay("blockgiro",1);
}

```

7.8.3 FASE 2 - ASP.NET, Silverlight

Código ASP.NET y JSP

```

<html>
<head>
<title>Catálogo 2008 dnp</title>
<!--Dream Weaver-->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script type="text/javascript">
<!--
function MM_swapImgRestore() { //v3.0
    var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_preloadImages() { //v3.0
    var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}

function MM_findObj(n, d) { //v4.01
    var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
        d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
    if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
    for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
    if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_swapImage() { //v3.0
    var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-
2);i+=3)
        if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src;
x.src=a[i+2];}
}
//-->
</script>

<!--Contenido Silverlight-->
<script type="text/javascript" src="Silverlight.js"></script>

```

```
<script type="text/javascript" src="Page.xaml.js"></script>
<style type="text/css">
#silverlightControlHost
{
height: 350px;
width: 605px;
}
#errorLocation
{
font-size: small;
color: Gray;
}
</style>
<script type="text/javascript">
function createSilverlight()
{
var scene = new Catálogo_dnp.Page();
Silverlight.createObjectEx({
source: "Page.xaml",
parentElement: document.getElementById("silverlightControlHost"),
id: "SilverlightControl",
properties:
{
width: "100%",
height: "100%",
version: "1.0"
},
events:
{
onLoad: Silverlight.createDelegate(scene, scene.handleLoad),
onError: function(sender, args) {
var errorDiv = document.getElementById("errorLocation");
if (errorDiv != null)
{
var errorText = args.errorType + "- " + args.errorMessage
if (args.ErrorType == "ParserError") {
errorText += "<br>File: " + args.xamlFile;
errorText += ", line " + args.lineNumber;

```

```

        errorText += " character " + args.charPosition;
    }
    else if (args.ErrorType == "RuntimeError") {
        errorText += "<br>line " + args.lineNumber;
        errorText += " character " + args.charPosition;
    }
    errorDiv.innerHTML = errorText;
}
}
}
});
}

if (!window.Silverlight)
    Silverlight = {};
    Silverlight.createDelegate = function(instance, method) {
        return function() {
            return method.apply(instance, arguments);
        }
    }

</script>
</head>

<body bgcolor="#FFFFFF" leftmargin="0" topmargin="0" marginwidth="0" marginheight="0"
onLoad="MM_preloadImages('images/master-master_04.jpg','images/master-
master_05.jpg','images/master-master_06.jpg','images/master-master_07.jpg','images/master-
master_08.jpg','images/master-master_09.jpg')">
<!-- ImageReady Slices (interface master page.psd) -->
<p align="center">
    <!-- End ImageReady Slices -->
</p>
<p align="center">&nbsp;</p>
<div align="center">
    <table width="648" height="334" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td width="351"><div align="center">
                <table width="0" border="0" cellpadding="0" cellspacing="0">
                    <tr>
                        <td>


```

```

</td>
    </tr>
    <tr>
        <td>
</td>
        </tr>
    </table>
</div></td>
    <td width="291"><div align="center">
        <table width="0" border="0" cellpadding="0" cellspacing="0">
            <tr>
                <td>
<div align="center">

</div></td>
            </tr>
            <tr>
                <td><div align="center">
                    <table width="0" border="0" cellpadding="0" cellspacing="0">
                        <tr>
                            <td></td>
                            <td><div align="center"><a href="#" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('HOME','','images/master-master_04.jpg',1)"></a></div></td>
                        <tr>
                            <td><div align="center"><a href="#" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('dnp k!ds','','images/master-master_05.jpg',1)"></a></div></td>
                            <td><div align="center"><a href="#" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('MUJER','','images/master-master_06.jpg',1)"></a></div></td>
                            <td><div align="center"><a href="#" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('hombre','','images/master-master_07.jpg',1)"></a></div></td>

```

```

<td><div align="center"><a href="#" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('Uniformes','','images/master-master_08.jpg',1)">
</a></div></td>

<td>

<div align="center">
<a href="#" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('Registro','','images/master-master_09.jpg',1)">
</a></div></td>

<td></td>

</tr>

</table>

</div></td>

</tr>

<tr>

<td><div align="center">

<!--contenido silverligh-->
<div id="silverlightControlHost">
<script type="text/javascript">
createSilverlight();
</script>
</div>

<!-- Runtime errors from Silverlight will be displayed here.
This will contain debugging information and should be removed or hidden when
debugging is completed -->
<div id='errorLocation'></div>

</div></td>

</tr>

<tr>

<td></td>

</tr>

</table>

```

```

        </div></td>
    </tr>
</table>
</div>
<div align="center"></div>
<p align="center">&nbsp;</p>
</body>
</html>

```

Código XAML

```

<Canvas
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="605" Height="350"
    Background="White"
    x:Name="Page">
    <Rectangle Width="605" Height="350" Fill="#FF131415" Stroke="#FF131415"/>
    <MediaElement Width="520" Height="334" Canvas.Left="40" Canvas.Top="8"
Source="Videos/Silverlight_Dusk_Widescreen_dreamscene.wmv"/>
</Canvas>

```

7.8.4 FASE 3 – Store Procedure SQL (MSQL Express)

Se ejecuta “Store Procedure” empleando SQL como se muestra a continuación:

Código SQL

```

Create Procedure SP_ValidarUsuario
@usu_UsuarioNombre varchar (40),
@usu_contrasena nchar(10)

AS
Begin
declare @Resultado numeric(1);
declare @usuario varchar(40);
declare @contrasena nchar(10);

SET @usuario =(SELECT [usu_UsuarioNombre] FROM
[Catálogo_DNP].[dbo].[Usuario]);
SET @contrasena = (SELECT [usu_contrasena] FROM
[Catálogo_DNP].[dbo].[Usuario]);

IF
        @usu_UsuarioNombre=@usuario
    SET @Resultado = '1'
else

```

```
        SET @Resultado = '0'  
END  
        select @Resultado  
Go
```

Finalmente se retorna el resultado a la aplicación en ASP.Net que continúa el proceso de carga de sistema de Catálogo según el resultado obtenido

```
execute SP_ValidarUsuario 'usuario','contraseña'
```

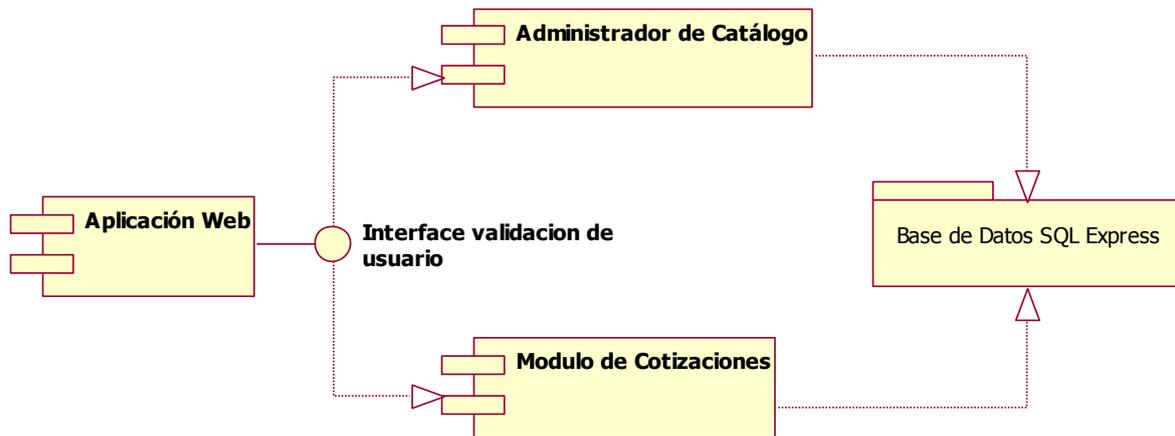
7.8. Anexo “H”: Diagrama De Componentes

DIAGRAMA DE COMPONENTES

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight(WPF/E) y XAML”

Versión 1.2.0

Diagrama De Componentes



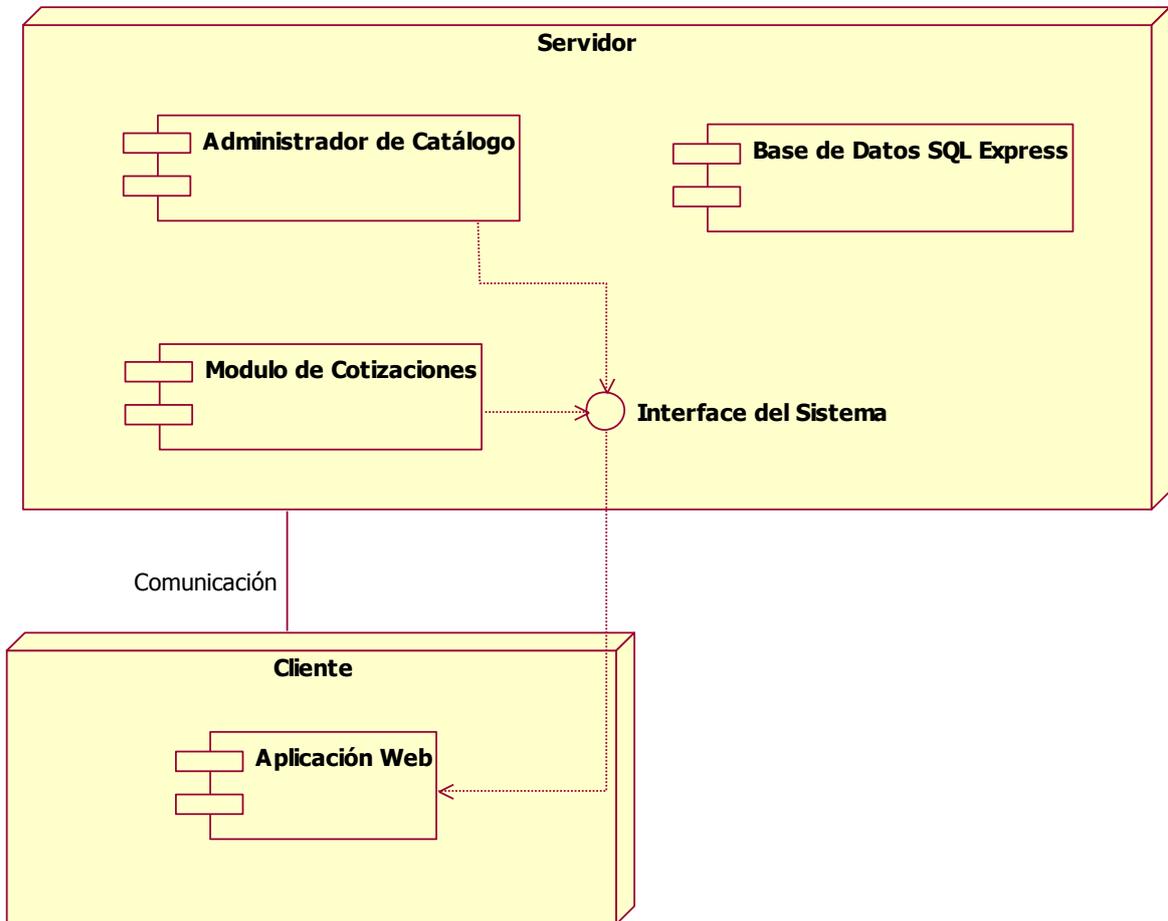
7.9. Anexo “I”: Diagrama De Despliegue

DIAGRAMA DE DESPLIEGUE

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight(WPF/E) y XAML”

Versión 1.2.0

Diagrama De Despliegue



7.10. Anexo "J": Resumen De Casos De Pruebas

RESUMEN DE CASOS DE PRUEBA

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight (WPF/E) y XAML”

Versión 1.2.0

7.10.1 Caso de Uso: Gestionar Cotizaciones

Introducción

Este documento especifica las funciones del Sistema de Catálogo propuesto, para realizar las pruebas que garanticen un sistema que satisfaga o supere las necesidades de la empresa en la que se implementará el sistema.

Propósito

El propósito de este documento de casos de prueba, es determinar una guía que oriente las pruebas a realizarse en cuanto a gestionar cotizaciones.

Alcance

En este documento identificaremos todos los casos de pruebas del caso de uso Gestionar cotizaciones con los cuales se seguirá las pruebas en el sistema.

Casos de prueba para requerimientos funcionales

Identificación de flujos de eventos

Flujo	Nombre	Detalle
Gestionar Cotizaciones	FB	Flujo Básico
Registrar Nuevo Usuario	FA1	Flujo Alternativo
No ingresa sus datos	FE1	Flujo de Excepción
Ingreso Incorrectamente su id de usuario y contraseña	FA2	Flujo Alternativo
Confirma su respuesta en desacuerdo	FA3	Flujo Alternativo
Usuario Sale de la Aplicación	FE2	Flujo de Excepción
Decide continuar con una nueva cotización	FA4	Flujo Alternativo

Identificación de los escenarios de prueba

Escenarios de Prueba	Detalle	FB	FA1	FE1	FA2	FA3	FE2	FA4
Escenario 1	Usuario registrado; ingresa sus datos de manera correcta	X						
Escenario 2	Usuario nuevo desea registrar sus datos	X	X					
Escenario 3	Usuario no ingresa sus datos	X		X				
Escenario 4	Usuario registrado; ingresa sus datos de manera incorrecta	X			X			
Escenario 5	Usuario registrado; recibe un resultado y confirma su respuesta en desacuerdo	X				X		
Escenario 6	Usuario Registrado recibe un resultado de cotización pero sale de la aplicación						X	
Escenario 7	Usuario Sale del Módulo de Cotización	X						X

Resumen de casos de prueba

Test Case ID#	Escenarios de Prueba	Condición	Precondiciones	Pos condiciones
TC1	Escenario 1	Usuario registrado; ingresa sus datos de manera correcta	Datos ingresados validos	La gestión de cotizaciones se da normalmente
TC2	Escenario 2	Usuario nuevo desea registrar sus datos	Datos personales de nuevo usuario existentes	Transferencia a interface de usuario para registro
TC3	Escenario 3	Usuario no ingresa sus datos	Inexistencia de datos	Desaplique de mensaje de error en Interface de Usuario de ingreso de clientes registrados y solicitud de nuevo ingreso
TC4	Escenario 4	Usuario registrado; ingresa sus datos de manera incorrecta	Datos Ingresados inválidos	Desaplique de mensaje de error en Interface de Usuario de ingreso de clientes registrados y solicitud de nuevo ingreso
TC5	Escenario 5	Usuario registrado; recibe un resultado y confirma su respuesta en desacuerdo	Selección exitosa de parámetros definidos en formulario de consulta de prendas	Devolución de resultado por parte del sistema
TC6	Escenario 6	Usuario Registrado recibe un resultado de cotización pero sale de la aplicación	Selección exitosa de parámetros definidos en formulario de consulta de prendas	Devolución de resultado por parte del sistema
TC7	Escenario 7	Usuario Sale del Módulo de Cotización	selección de cierre de aplicación	cierre de aplicación

Definición de datos de prueba

En esta sección se definirán los campos de datos, así como también los posibles valores que se van a utilizar en los casos de prueba correspondientes al caso de uso.

Datos Usuario

nombre	Juan
contraseña	jamiroquai

Datos Formulario de Cotización

Prenda	Calentador
Modelo	Opción 2
Tela	Topper
Talla	34
Color Primario	Azul
Color Secundario	Marino
Cantidad de Uniformes	12
Extras	2 Rib/4 Puños/ 2 cierres
Estampado	2 Monocromático/Pequeño

Gestionar cotizaciones

Test Case ID#	Escenarios de Prueba	Condición	Valido/Invalido	N. Usuario	Contraseña	Resultados Esperados	Resultados Dados
TC1	Escenario 1	Usuario registrado; ingresa sus datos de manera correcta	V	Juan	jamiroquai	Resultado de cotización para cliente/ Registro en sistema de cotización efectuada	Resultado de cotización para cliente/ Registro en sistema de cotización efectuada
TC2	Escenario 2	Usuario nuevo desea registrar sus datos	V	Juan	jamiroquai	Se transfiere al usuario a formulario de nuevo usuario	Se transfiere al usuario a formulario de nuevo usuario
TC3	Escenario 3	Usuario no ingresa sus datos	I			el sistema no ingresa a gestión de cotizaciones	el sistema no ingresa a gestión de cotizaciones
TC4	Escenario 4	Usuario registrado; ingresa sus datos de manera incorrecta	I	Juan	jk	el sistema no ingresa a gestión de cotizaciones	el sistema no ingresa a gestión de cotizaciones
TC5	Escenario 5	Usuario registrado; recibe un resultado y confirma su respuesta en desacuerdo	V	Juan	jamiroquai	Resultado de cotización para cliente/ Registro en sistema de cotización efectuada	Resultado de cotización para cliente/ Registro en sistema de cotización efectuada
TC6	Escenario 6	Usuario Registrado recibe un resultado de cotización pero sale de la aplicación	V	Juan	jamiroquai	Resultado de cotización para cliente/ No se da un registro en sistema de cotización efectuada	Resultado de cotización para cliente/ No se da un registro en sistema de cotización efectuada
TC7	Escenario 7	Usuario Sale del Módulo de Cotización	V	Juan	jamiroquai	Resultado de cotización para cliente/ Registro en sistema de cotización efectuada y termina Gestión de Cotizaciones	Resultado de cotización para cliente/ Registro en sistema de cotización efectuada y termina Gestión de Cotizaciones

7.10.2 Caso de Uso: Registrar Usuario

Introducción

Este documento especifica las funciones del Sistema de Catálogo propuesto, para realizar las pruebas que garanticen un sistema que satisfaga o supere las necesidades de la empresa en la que se implementará el sistema.

Propósito

El propósito de este documento de casos de prueba, es determinar una guía que oriente las pruebas a realizarse en cuanto a registrar el usuario.

Alcance

En este documento identificaremos todos los casos de pruebas del caso de uso Registrar Usuario con los cuales se seguirá las pruebas en el sistema.

Casos de prueba para requerimientos funcionales

Identificación de flujos de eventos

Flujo	Nombre	Detalle
Registrar Usuario	FB	Flujo Básico
Usuario Cancela Ingreso	FE1	Flujo de Excepción
Usuario no envía datos	FE2	Flujo de Excepción
Usuario no envió la información completa o correcta	FA1	Flujo Alternativo

Identificación de los escenarios de prueba

Escenarios de Prueba	Detalle	FB	FE1	FA2	FA1
Escenario 1	Usuario registrado; ingresa sus datos de manera correcta	X			
Escenario 2	Usuario Cancela ingreso de datos	X	X		
Escenario 3	Usuario no envía datos	X		X	
Escenario 4	Usuario no envía información Completa	X			X

Resumen de casos de prueba

Test Case ID#	Escenarios de Prueba	Condición	Precondiciones	Pos condiciones
TC1	Escenario 1	Usuario registrado; ingresa sus datos de manera correcta	Datos ingresados validos	Registro de usuario Exitoso
TC2	Escenario 2	Usuario Cancela ingreso de datos	No se ingresan datos	Transferencia a interface de usuario para registro
TC3	Escenario 3	Usuario no envía datos	Inexistencia de datos	Desaplique de mensaje de error en Interface de Usuario de ingreso de clientes registrados y solicitud de nuevo ingreso
TC4	Escenario 4	Usuario no envía información Completa	Datos Ingresados inválidos o incompletos	Desaplique de mensaje de error en Interface de Usuario de registro de usuario con solicitud de corrección en ingreso

Definición de datos de prueba

En esta sección se definirán los campos de datos, así como también los posibles valores que se van a utilizar en los casos de prueba correspondientes al caso de uso.

Datos personales de usuario

Descripción	Valor	Tipo
Id usuario	2	identity
Nombre	Silvia	requerido
Apellido	Ascázubi	requerido
Teléfono	2864XXX	requerido
Celular	96123XXX	opcional
dirección	Av. Río Pastaza 10X	requerido
ciudad	Quito	requerido
institución		opcional
contraseña	ivita	requerido

Registrar Usuario

Test Case ID#	Escenarios de Prueba	Condición	Valido/Invalido	Id usuario	Nombre	Apellido	Teléfono	Celular	dirección	ciudad	institución	contraseña	Resultados Esperados	Resultados Dados
TC1	Escenario 1	Usuario registrado; ingresa sus datos de manera correcta	V	2	Silvia	Ascázubi	2864675	961235	Av. Río Pastaza 1015	Quito		ivita	Se registra al nuevo usuario en la base de datos del sistema	Se registra al nuevo usuario en la base de datos del sistema
TC2	Escenario 2	Usuario Cancela ingreso de datos	V										se termina el ingreso de usuario sin registrarlo	se termina el ingreso de usuario sin registrarlo
TC3	Escenario 3	Usuario no envía datos	I										Se solicita reingreso de datos que la operación no se completo	Se solicita reingreso de datos que la operación no se completo
TC4	Escenario 4	Usuario no envía información Completa	I	2	Silvia		xyz						Se solicita reingreso de datos que la operación no se completo	Se solicita reingreso de datos que la operación no se completo

7.10.3 Caso de Uso: Consultar Catálogo

Introducción

Este documento especifica las funciones del Sistema de Catálogo propuesto, para realizar las pruebas que garanticen un sistema que satisfaga o supere las necesidades de la empresa en la que se implementará el sistema.

Propósito

El propósito de este documento de casos de prueba, es determinar una guía que oriente las pruebas a realizarse en cuanto a consultar el catálogo.

Alcance

En este documento identificaremos todos los casos de pruebas del caso de uso Consultar Catálogo con los cuales se seguirá las pruebas en el sistema.

Casos de prueba para requerimientos funcionales

Identificación de flujos de eventos

Flujo	Nombre	Detalle
Consultar catálogo	FB	Flujo Básico
Usuario Cancela consulta	FE1	Flujo de Excepción
No existen Productos que cumplan criterio de búsqueda	FA1	Flujo Alterno

Identificación de los escenarios de prueba

Escenarios de Prueba	Detalle	FB	FE1	FA1
Escenario 1	Usuario consulta catálogo de productos según categorías disponibles	X		
Escenario 2	Usuario Cancela consulta a catálogo	X	X	
Escenario 3	El sistema no despliega productos disponibles	X		X

Resumen de casos de prueba

Test Case ID#	Escenarios de Prueba	Condición	Precondiciones	Pos condiciones
TC1	Escenario 1	Usuario consulta catálogo de productos según categorías disponibles	Existe disponibilidad de productos	Consulta exitosa
TC2	Escenario 2	Usuario Cancela consulta a catálogo	Se cancela la consulta	Se cancela la consulta
TC3	Escenario 3	El sistema no despliega productos disponibles	Inexistencia de Productos o categorías	Inexistencia de Productos o categorías

Definición de datos de prueba

En esta sección se definirán los campos de datos, así como también los posibles valores que se van a utilizar en los casos de prueba correspondientes al caso de uso.

Consulta de prendas

Prenda	Calentador
Modelo	Opción 2
Tela	Topper
Talla	34
Color Primario	Azul
Color Secundario	Marino
Cantidad de Uniformes	12
Extras	2 Rib/4 Puños/ 2 cierres
Estampado	2 Monocromático/Pequeño

Consultar Catálogo

Test Case ID#	Escenarios de Prueba	Condición	Valido/Invalido	Prenda	Modelo	Tela	Talla	Color Primario	Color Secundario	Cantidad de Uniformes	Extras	Estampado	Resultados Esperados	Resultados Dado
TC1	Escenario 1	Usuario consulta catálogo de productos según categorías disponibles	V	Calentador	Opción 2	Topper	34	Azul	Marino	12	2 Rib/4 Puños/2 cierres	2 Monocromático/Pequeno	Consulta exitosa en base a parámetros escogidos por el usuario	Consulta exitosa en base a parámetros escogidos por el usuario
TC2	Escenario 2	Usuario Cancela consulta a catálogo	V										Termina operación de consulta	Termina operación de consulta
TC3	Escenario 3	El sistema no despliega productos disponibles	V	no está disponible	no está disponible	no está disponible	Consulta exitosa en base a parámetros escogidos por el usuario	Inexistencia de productos solicitados						

7.10.4 Caso de Uso: Validar Usuario

Introducción

Este documento especifica las funciones del Sistema de Catálogo propuesto, para realizar las pruebas que garanticen un sistema que satisfaga o supere las necesidades de la empresa en la que se implementará el sistema.

Propósito

El propósito de este documento de casos de prueba, es determinar una guía que oriente las pruebas a realizarse en cuanto a validar al usuario.

Alcance

En este documento identificaremos todos los casos de pruebas del caso de uso Validar Usuario con los cuales se seguirá las pruebas en el sistema.

Casos de prueba para requerimientos funcionales

Identificación de flujos de eventos

Flujo	Nombre	Detalle
Validar Usuario	FB	Flujo Básico

Identificación de los escenarios de prueba

Escenarios de Prueba	Detalle	FB
Escenario 1	El sistema procesa los datos ingresados por el usuario y los valida retornando un resultado a favor o en desacuerdo según la información almacenada en la base de datos del sistema	X

Resumen de casos de prueba

Test Case ID#	Escenarios de Prueba	Condición	Precondiciones	Pos condiciones
TC1	Escenario 1	El sistema procesa los datos ingresados por el usuario y los valida retornando un resultado a favor o en desacuerdo según la información almacenada en la base de datos del sistema	Se recibe parámetros a validar de usuario	Consulta exitosa

Definición de datos de prueba

En esta sección se definirán los campos de datos, así como también los posibles valores que se van a utilizar en los casos de prueba correspondientes al caso de uso.

Datos Usuario

nombre	Juan
--------	------

Validar Usuario

Test Case ID#	Escenarios de Prueba	Condición	Valido/Invalido	N. Usuario	Contraseña	Resultados Esperados	Resultados Dados
TC1	Escenario 1	El sistema procesa los datos ingresados por el usuario y los valida retornando un resultado a favor o en desacuerdo según la información almacenada en la base de datos del sistema	V	Juan	jamiroquai	Validación de consulta de usuario	Consulta exitosa el usuario esta registrado

7.10.5 Caso de Uso: Gestionar Categorías

Introducción

Este documento especifica las funciones del Sistema de Catálogo propuesto, para realizar las pruebas que garanticen un sistema que satisfaga o supere las necesidades de la empresa en la que se implementará el sistema.

Propósito

El propósito de este documento de casos de prueba, es determinar una guía que oriente las pruebas a realizarse en cuanto a gestionar categorías.

Alcance

En este documento identificaremos todos los casos de pruebas del caso de uso Gestionar Categorías con los cuales se seguirá las pruebas en el sistema.

Casos de prueba para requerimientos funcionales

Identificación de flujos de eventos

Flujo	Nombre	Detalle
Gestionar Categorías	FB	Flujo Básico
Ingresar nueva categoría	SF1	Sub flujo
Modificar Categoría	SF2	Sub flujo
Eliminar Categoría	SF3	Sub flujo

Identificación de los escenarios de prueba

Escenarios de Prueba	Detalle	FB	SF1	SF2	SF3
Escenario 1	Webmaster gestiona categorías	X			
Escenario 2	Webmaster ingresa categorías	X	X		
Escenario 3	Webmaster modifica categoría	X		X	
Escenario 4	Webmaster elimina categoría	X			X

Resumen de casos de prueba

Test Case ID#	Escenarios de Prueba	Condición	Precondiciones	Pos condiciones
TC1	Escenario 1	Webmaster gestiona categorías	Ingresa gestión de categorías	Gestiona Categorías de forma exitosa
TC2	Escenario 2	Webmaster ingresa categorías	Las categorías a ingresar están revisadas y listas para ser anexadas a la aplicación	Se ingresa exitosamente las categorías
TC3	Escenario 3	Webmaster modifica categoría	Se tiene acceso a colección de categorías existentes	Se modifica exitosamente las categorías
TC4	Escenario 4	Webmaster elimina categoría	Se tiene acceso a colección de categorías existentes	Se elimina exitosamente las categorías

Definición de datos de prueba

En esta sección se definirán los campos de datos, así como también los posibles valores que se van a utilizar en los casos de prueba correspondientes al caso de uso.

Categoría

Descripción	Valor
categoría	uniformes
su categoría	catering

Gestionar Categorías

Test Case ID#	Escenarios de Prueba	Condición	Valido/Invalido	categoría	sub categoría	Resultados Esperados	Resultados Dados
TC1	Escenario 1	Webmaster gestiona categorías	V	uniformes	catering	Gestiona Categorías de forma exitosa	Gestiona Categorías de forma exitosa
TC2	Escenario 2	Webmaster ingresa categorías	V	uniformes	catering	Se ingresa exitosamente las categorías	Se ingresa exitosamente las categorías
TC3	Escenario 3	Webmaster modifica categoría	V	uniformes	catering	actualización de categorías	Se modifica exitosamente las categorías
TC4	Escenario 4	Webmaster elimina categoría	V	uniformes	catering	actualización de categorías	Se elimina exitosamente las categorías

7.10.6 Caso de Uso: Gestionar Imágenes

Introducción

Este documento especifica las funciones del Sistema de Catálogo propuesto, para realizar las pruebas que garanticen un sistema que satisfaga o supere las necesidades de la empresa en la que se implementará el sistema.

Propósito

El propósito de este documento de casos de prueba, es determinar una guía que oriente las pruebas a realizarse en cuanto a gestionar imágenes.

Alcance

En este documento identificaremos todos los casos de pruebas del caso de uso Gestionar imágenes con los cuales se seguirá las pruebas en el sistema.

Casos de prueba para requerimientos funcionales

Identificación de flujos de eventos

Flujo	Nombre	Detalle
Gestionar Imágenes	FB	Flujo Básico
Ingresar nueva Imagen	SF1	Sub flujo
Modifica Imagen	SF2	Sub flujo
Elimina Imagen	SF3	Sub flujo

Identificación de los escenarios de prueba

Escenarios de Prueba	Detalle	FB	SF1	SF2	SF3
Escenario 1	Webmaster gestiona Imágenes	X			
Escenario 2	Webmaster ingresa Imagen	X	X		
Escenario 3	Webmaster modifica Imagen	X		X	
Escenario 4	Webmaster elimina Imagen	X			X

Resumen de casos de prueba

Test Case ID#	Escenarios de Prueba	Condición	Precondiciones	Pos condiciones
TC1	Escenario 1	Webmaster gestiona categorías	Ingresa gestión de Imágenes	Gestiona imágenes de forma exitosa
TC2	Escenario 2	Webmaster ingresa Imagen	Las imágenes a ingresar están revisadas y listas para ser anexadas a la aplicación	Se ingresa exitosamente las imágenes
TC3	Escenario 3	Webmaster modifica Imagen	Se tiene acceso a colección de imágenes existentes	Se modifica exitosamente las imágenes
TC4	Escenario 4	Webmaster elimina Imagen	Se tiene acceso a colección de imágenes existentes	Se elimina exitosamente las imágenes

Definición de datos de prueba

En esta sección se definirán los campos de datos, así como también los posibles valores que se van a utilizar en los casos de prueba correspondientes al caso de uso.

Imagen

Descripción	Valor
imagen	img34
tipo	JPEG

Gestionar Imágenes

Test Case ID#	Escenarios de Prueba	Condición	Valido/Invalido	categoría	sub categoría	Resultados Esperados	Resultados Dados
TC1	Escenario 1	Webmaster gestiona imágenes	V	img34	JPEG	Gestiona Imágenes de forma exitosa	Gestiona Imágenes de forma exitosa
TC2	Escenario 2	Webmaster ingresa Imagen	V	img34	JPEG	Se ingresa exitosamente las Imágenes	Se ingresa exitosamente las Imágenes
TC3	Escenario 3	Webmaster modifica Imagen	V	img34	JPEG	actualización de Imágenes	Se modifica exitosamente las Imágenes
TC4	Escenario 4	Webmaster elimina Imagen	V	img34	JPEG	actualización de Imágenes	Se elimina exitosamente las Imágenes

7.10.7 Caso de Uso: Gestionar Productos

Introducción

Este documento especifica las funciones del Sistema de Catálogo propuesto, para realizar las pruebas que garanticen un sistema que satisfaga o supere las necesidades de la empresa en la que se implementará el sistema.

Propósito

El propósito de este documento de casos de prueba, es determinar una guía que oriente las pruebas a realizarse en cuanto a gestionar productos.

Alcance

En este documento identificaremos todos los casos de pruebas del caso de uso Gestionar Productos con los cuales se seguirá las pruebas en el sistema.

Casos de prueba para requerimientos funcionales

Identificación de flujos de eventos

Flujo	Nombre	Detalle
Gestionar Productos	FB	Flujo Básico
Ingresa nuevo Producto	SF1	Sub flujo
Modifica Producto	SF2	Sub flujo
Elimina Producto	SF3	Sub flujo

Identificación de los escenarios de prueba

Escenarios de Prueba	Detalle	FB	SF1	SF2	SF3
Escenario 1	Webmaster gestiona Productos	X			
Escenario 2	Webmaster ingresa Producto	X	X		
Escenario 3	Webmaster modifica Producto	X		X	
Escenario 4	Webmaster elimina Producto	X			X

Resumen de casos de prueba

Test Case ID#	Escenarios de Prueba	Condición	Precondiciones	Pos condiciones
TC1	Escenario 1	Webmaster gestiona Productos	Ingresa gestión de Productos	Gestiona Productos de forma exitosa
TC2	Escenario 2	Webmaster ingresa Producto	Los Productos a ingresar están aprobados y listos para ser anexadas a la aplicación	Se ingresa exitosamente los Productos
TC3	Escenario 3	Webmaster modifica Producto	Se tiene acceso a colección de Productos existentes	Se modifica exitosamente los Productos
TC4	Escenario 4	Webmaster elimina Producto	Se tiene acceso a colección de Productos existentes	Se elimina exitosamente los Productos

Definición de datos de prueba

En esta sección se definirán los campos de datos, así como también los posibles valores que se van a utilizar en los casos de prueba correspondientes al caso de uso.

Producto

Prenda	Calentador
Modelo	Opción 2
Tela	Topper
Talla	34
Color Primario	Azul
Color Secundario	Marino
Extras	2 Rib/4 Puños/ 2 cierres
Estampado	2 Monocromático/Pequeño

Gestionar Producto

Test Case ID#	Escenarios de Prueba	Condición	Valido/Invalido	Prenda	Modelo	Tela	Talla	Color Primario	Color Secundario	Extras	Estampado	Resultados Esperados	Resultados Dados
TC1	Escenario 1	Webmaster gestiona categorías	V									Gestiona Productos de forma exitosa	Gestiona Productos de forma exitosa
TC2	Escenario 2	Webmaster ingresa Producto	V	Calentador	Opción 2	Topper	34	Azul	Marino	2 Rib/4 Puños/2 cierres	2 Monocromático/Pequeño	Se ingresa exitosamente Producto y descripción	Se ingresa exitosamente Productos
TC3	Escenario 3	Webmaster modifica Producto	V	Calentador	Opción 2	Topper	34	Azul	Marino	2 Rib/4 Puños/2 cierres	2 Monocromático/Pequeño	actualización de Productos	Se modifica exitosamente Productos
TC4	Escenario 4	Webmaster elimina Producto	V	Calentador	Opción 2	Topper	34	Azul	Marino	2 Rib/4 Puños/2 cierres	2 Monocromático/Pequeño	actualización de Productos	Se elimina exitosamente Productos

7.10.8 Caso de Uso: Gestionar Novedades

Introducción

Este documento especifica las funciones del Sistema de Catálogo propuesto, para realizar las pruebas que garanticen un sistema que satisfaga o supere las necesidades de la empresa en la que se implementará el sistema.

Propósito

El propósito de este documento de casos de prueba, es determinar una guía que oriente las pruebas a realizarse en cuanto al cambio de estado de sesión del usuario.

Alcance

En este documento identificaremos todos los casos de pruebas del caso de uso Gestionar Novedades con los cuales se seguirá las pruebas en el sistema.

Casos de prueba para requerimientos funcionales

Identificación de flujos de eventos

Flujo	Nombre	Detalle
Gestionar Novedades	FB	Flujo Básico
Ingresa nueva Novedad	SF1	Sub flujo
Modifica Novedad	SF2	Sub flujo
Elimina Novedad	SF3	Sub flujo

Identificación de los escenarios de prueba

Escenarios de Prueba	Detalle	FB	SF1	SF2	SF3
Escenario 1	Webmaster gestiona Novedades	X			
Escenario 2	Webmaster ingresa Novedad	X	X		
Escenario 3	Webmaster modifica Novedad	X		X	
Escenario 4	Webmaster elimina Novedad	X			X

Resumen de casos de prueba

Test Case ID#	Escenarios de Prueba	Condición	Precondiciones	Pos condiciones
TC1	Escenario 1	Webmaster gestiona categorías	Ingresa gestión de Novedades	Gestiona Novedades de forma exitosa
TC2	Escenario 2	Webmaster ingresa Novedad	Las Novedades a ingresar están revisadas y listas para ser anexadas a la aplicación	Se ingresa exitosamente las Novedades
TC3	Escenario 3	Webmaster modifica Novedad	Se tiene acceso a colección de Novedades existentes	Se modifica exitosamente las Novedades
TC4	Escenario 4	Webmaster elimina Novedad	Se tiene acceso a colección de Novedades existentes	Se elimina exitosamente las Novedades

Definición de datos de prueba

En esta sección se definirán los campos de datos, así como también los posibles valores que se van a utilizar en los casos de prueba correspondientes al caso de uso.

Archivo de novedad

Descripción	Valor
Novedad	Moda verano 2008
Archivo	Temporada.xml

Gestionar Novedad

Test Case ID#	Escenarios de Prueba	Condición	Valido/Invalido	Novedad	Archivo	Resultados Esperados	Resultados Dados
TC1	Escenario 1	Webmaster gestiona categorías	V			Gestiona Novedades de forma exitosa	Gestiona Novedades de forma exitosa
TC2	Escenario 2	Webmaster ingresa Novedad	V	Moda verano 2008	Temporada.xml	Se ingresa exitosamente las Novedades	Se ingresa exitosamente las Novedades
TC3	Escenario 3	Webmaster modifica Novedad	V	Moda verano 2008	Temporada.xml	actualización de Novedades	Se modifica exitosamente las Novedades
TC4	Escenario 4	Webmaster elimina Novedad	V	Moda verano 2008	Temporada.xml	actualización de Novedades	Se elimina exitosamente las Novedades

7.11. Anexo “K”: Glosario de Términos

GLOSARIO DE TÉRMINOS

“Sistema de Cotización de Uniformes mediante catálogo en línea empleando Silverlight(WPF/E) y XAML”

Versión 1.2.0

Introducción

En este documento se incluyen todos términos que necesiten explicación adicional, con el objeto de que el usuario tenga un superior entendimiento del sistema a construir.

Propósito

El propósito de este documento es proporcionar información sobre los términos importantes utilizados en el desarrollo del sistema con el fin de que las personas involucradas con el desarrollo del sistema tengan una mejor concepción del mismo.

Alcance

Este glosario de datos incluye los términos más importantes relacionados con el sistema a ser desarrollado.

Glosario

Browser

Código	G01
Nombre del término	Navegador
Nombre del término en inglés	Browser
Descripción	Programa usado para visualizar documentos WWW en Internet. Los más utilizados en estos momentos son Netscape e Internet Explorer

Hipertexto

Código	G02
Nombre del término	Hipertexto
Nombre del término en inglés	Hypertext
Descripción	Es el nombre que recibe el texto que en la pantalla de una computadora conduce a su usuario a otro texto relacionado. La forma más habitual de hipertexto en documentos es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos (lexias). Si el usuario selecciona un hipervínculo, hace que el programa de la computadora muestre inmediatamente el documento enlazado

HTML

Código	G03
Nombre del término	Lenguaje de Marcado de Hipertexto
Nombre del término en inglés	HyperText Markup Language
Descripción	Es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Protocolo HTTP

Código	G04
Nombre del término	HTTP
Nombre del término en inglés	HyperText Transfer Protocol
Descripción	Es el protocolo usado en cada transacción de la Web (WWW). HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden.

Ingeniería de Software

Código	G05
Nombre del término	Ingeniería de Software
Nombre del término en inglés	
Descripción	La Ingeniería de software designa el conjunto de técnicas destinadas a la producción de un programa de computadora, más allá de la sola actividad de programación. Forman parte de esta

	disciplina las ciencias computacionales y el manejo de proyectos, entre otros campos, propios de la rama más genérica denominada Ingeniería informática
--	---

Internet

Código	G06
Nombre del término	Internet
Nombre del término en inglés	
Descripción	es un método de interconexión descentralizada de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP y garantiza que redes físicas heterogéneas funcionen como una red lógica única, de alcance mundial

Intranet

Código	G07
Nombre del término	Intranet
Nombre del término en inglés	Intranet
Descripción	Una intranet es una red de ordenadores de una red de área local (LAN) privada empresarial o educativa que proporciona herramientas de Internet, la cual tiene como función principal proveer lógica de negocios para aplicaciones de captura, reportes, consultas, etc.

Multiplataforma

Código	G08
Nombre del término	Multiplataforma
Nombre del término en inglés	
Descripción	Multiplataforma es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas

Página Web

Código	G09
Nombre del término	Sitio Web
Nombre del término en inglés	Web Site
Descripción	Es una fuente de información adaptada para la World Wide Web (WWW) y accesible mediante un navegador de Internet. Esta información se presenta generalmente en formato HTML y puede contener hiperenlaces a otras páginas Web, constituyendo la red enlazada de la World Wide Web.

Plug - In

Código	G11
Nombre del término	Enchufe, aditamento
Nombre del término en inglés	Plug In

Descripción	También conocido como addin, add-in, addon o add-on) es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica, como por ejemplo servir como driver (controlador) en una aplicación, para hacer así funcionar un dispositivo en otro programa. Ésta aplicación adicional es ejecutada por la aplicación principal. Los plugins típicos tienen la función de reproducir determinados formatos de gráficos, reproducir datos multimedia, codificar/decodificar emails, filtrar imágenes de programas gráficos.
-------------	---

Portales

Código	G12
Nombre del término	Portales
Nombre del término en inglés	
Descripción	Un portal de Internet es un sitio Web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, documentos, aplicaciones, compra electrónica, etc.

Protocolo de Red

Código	G13
Nombre del término	Protocolo de Red
Nombre del término en inglés	
Descripción	Protocolo de red o también Protocolo de Comunicación es el conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre las entidades que forman parte de una red.

Protocolo FTP

Código	G14
Nombre del término	FTP
Nombre del término en inglés	File Transfer Protocol
Descripción	Es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.

Sitio Web

Código	G15
Nombre del término	Sitio Web
Nombre del término en inglés	Web Site
Descripción	es un conjunto de páginas Web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet

Servidor Web

Código	G16
Nombre del término	Servidor Web
Nombre del término en inglés	
Descripción	Un servidor Web es un programa que implementa el protocolo HTTP. Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador.

Software

Código	G17
Nombre del término	Software
Nombre del término en inglés	
Descripción	Se refiere al equipamiento lógico o soporte lógico de un computador digital, comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware)

SQL

Código	G18
Nombre del término	SQL
Nombre del término en inglés	Structured Query Language
Descripción	Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre la misma. Es un lenguaje de cuarta generación (4GL).

Webmaster

Código	G19
Nombre del término	Webmaster
Nombre del término en inglés	Webmaster
Descripción	Persona encargada de la administración del contenido de las páginas Web

WIKI

Código	G20
Nombre del término	WIKI
Nombre del término en inglés	WIKI
Descripción	Es un sitio Web cuyas páginas Web pueden ser editadas por múltiples lectores a través del navegador Web. Los usuarios pueden crear, modificar o borrar un mismo texto que comparten

World Wide Web

Código	G21
Nombre del término	World Wide Web
Nombre del término en inglés	World Wide Web
Descripción	es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet

XHTML

Código	G22
Nombre del término	XHTML
Nombre del término en inglés	Extensible Hypertext Markup Language
Descripción	Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas Web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML.

WPF

Código	WPF01
Nombre del término	WPF
Nombre del término en inglés	Windows Presentation Foundation
Descripción	es el subsistema de presentación unificado de Microsoft para Windows, expuesto a través de "WinFX"; modelo de código administrado de Windows Vista que extiende Microsoft .NET Framework

CLR

Código	WPF02
Nombre del término	CLR
Nombre del término en inglés	Common Language Runtime
Descripción	Este es el lenguaje insignia de .NET Framework (marco de trabajo .NET) y pretende reunir las ventajas de lenguajes como C, C++ y Visual Basic en uno solo. El CLR es el verdadero núcleo del framework de .NET, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo (W2k y W2003).

.NET

Código	WPF03
Nombre del término	.NET
Nombre del término en inglés	.NET
Descripción	es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

DOM

Código	WPF04
Nombre del término	DOM (Modelo en Objetos para la representación de Documentos)
Nombre del término en inglés	Document Object Model
Descripción	Abreviado DOM, es esencialmente un modelo computacional a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML. Su objetivo es ofrecer un modelo orientado a objetos para el tratamiento y manipulación en tiempo real (o en forma dinámica) a la vez que de manera estática de páginas de Internet.

Active X

Código	SL01
Nombre del término	ActiveX
Nombre del término en inglés	ActiveX
Descripción	ActiveX es una tecnología de Microsoft para el desarrollo de páginas dinámicas. Tiene presencia en la programación del lado del servidor y del lado del cliente, aunque existan diferencias en el uso en cada uno de esos dos casos.

API

Código	SL02
Nombre del término	Interfaz de Programación de Aplicaciones
Nombre del término en inglés	Application Programming Interface
Descripción	es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Grafica Vectorial

Código	SL03
Nombre del término	Imagen Vectorial
Nombre del término en inglés	
Descripción	Una imagen vectorial es una imagen digital formada por objetos geométricos independientes (segmentos, polígonos, arcos, etc.), cada uno de ellos definido por distintos atributos matemáticos de forma, de posición, de color, etc. Por ejemplo un círculo de color rojo quedaría definido por la posición de su centro, su radio, el grosor de línea y su color.

Mainframe

Código	SL04
Nombre del término	Computador Central o Servidor
Nombre del término en inglés	Mainframe
Descripción	Una computadora central o mainframe es una computadora grande, potente y costosa usada principalmente por una gran compañía para el procesamiento de una gran cantidad de datos; por ejemplo, para el procesamiento de transacciones bancarias.

OLE

Código	SL05
Nombre del término	OLE - sistema de objeto distribuido
Nombre del término en inglés	Object Linking and Embedding
Descripción	<p>es un sistema de objeto distribuido y un protocolo desarrollado por Microsoft</p> <p>OLE permite a un editor encargar a otro la elaboración de parte de un documento y posteriormente volverlo a importar. Por ejemplo, un sistema de publicación de escritorio puede enviar un poco de texto a un procesador de textos o una imagen a un editor de bitmap usando OLE. La ventaja principal de usar OLE, además de que el tamaño del archivo es menor, es la de poder crear un archivo principal. Se puede hacer una referencia a los datos de ese archivo, con lo que todo cambio posterior en el archivo principal se reflejará en el documento referenciado.</p> <p>Su uso principal es el manejo de documentos compuestos (compound documents), pero también puede ser usado para transferir datos entre aplicaciones diferentes usando arrastrar y soltar y operaciones del portapapeles (clipboard)</p>

Píxel

Código	SL06
Nombre del término	Píxel
Nombre del término en inglés	Píxel (acrónimo de picture element)
Descripción	<p>es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.</p>

POX

Código	SL07
Nombre del término	POX
Nombre del término en inglés	Plain Old XML (POX)
Descripción	<p>término usado para describir XML básico, algunas veces mezclados con otros, mezcla de especificaciones como Nombre del espacio XML</p>

Renderización

Código	SL08
Nombre del término	Renderización
Nombre del término en inglés	Render
Descripción	<p>Es el proceso de generar una imagen desde un modelo. Los medios por los que se puede hacer un renderizado van desde lápiz, pluma, plumones o pastel, hasta medios digitales en dos y tres dimensiones. La palabra renderización proviene del inglés render, y no existe un verbo con el mismo significado en español, por lo que es frecuente usar las expresiones renderizar o renderear.</p>

Rasterización

Código	SL09
Nombre del término	Rasterización
Nombre del término en inglés	Raster o Bitmap
Descripción	<p>Una imagen rasterizada, también llamada bitmap, imagen matricial o pixmap, es una estructura o archivo de datos que representa una rejilla rectangular de píxeles o puntos de color, denominada raster, que se puede visualizar en un monitor de ordenador, papel u otro dispositivo de representación.</p> <p>A las imágenes rasterizadas se las suele caracterizar técnicamente por su altura y anchura (en píxeles) y por su profundidad de color (en bits por píxel), que determina el número de colores distintos que se pueden almacenar en cada píxel, y por lo tanto, en gran medida, la calidad del color de la imagen.</p> <p>Los gráficos rasterizados se distinguen de los gráficos vectoriales en que estos últimos representan una imagen a través del uso de objetos geométricos como curvas de Bézier y polígonos, no del simple almacenamiento del color de cada píxel. El formato de imagen matricial está ampliamente extendido y es el que se suele emplear para tomar fotografías digitales y realizar capturas de vídeo. Para su obtención se usan dispositivos de conversión analógica-digital, tales como escáneres y cámaras digitales.</p>

Silverlight

Código	SL09
Nombre del término	Silverlight
Nombre del término en inglés	Silverlight
Descripción	<p>es un complemento para navegadores de Internet que le agrega algunas de las funciones del Windows Presentation Foundation como la reproducción de vídeos, gráficos vectoriales, animaciones y otros elementos. El objetivo es brindar una aplicación similar a la que da Adobe Flash Player, una aplicación desarrollada por Macromedia ahora perteneciente a Adobe.</p>

Windows Media

Código	SL10
Nombre del término	Reproductor Windows Media
Nombre del término en inglés	Windows Media Player
Descripción	<p>Es un reproductor multimedia creado por la empresa Microsoft. El WMP permite reproducir diversos formatos digitales: Audio CD, DVD-Video, DVD-Audio, WMA (Windows Media Audio), WMV (Windows Media Video), MP3, MPG y AVI, entre otros, siempre y cuando, se dispongan de los códecs.</p> <p>Incluye acceso a video en formato digital en servidores de pago. También da la posibilidad de traspasar canciones de un CD al disco duro de la computadora y viceversa. Además, busca en Internet los nombres de las canciones y álbumes, además muestra la carátula del disco del cual provienen dichas canciones.</p>

WYSIWYG

Código	SL11
Nombre del término	WYSIWYG
Nombre del término en inglés	What You See Is What You Get
Descripción	En inglés, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso. Se dice en contraposición a otros procesadores de texto, hoy en día poco frecuentes, en los que se escribía sobre una vista que no mostraba el formato del texto, hasta la impresión del documento. En el caso de editores de HTML este concepto se aplica a los que permiten escribir la página sobre una vista preliminar similar a la de un procesador de textos, ocupándose en este caso el programa de generar el código fuente en HTML.

XAML

Código	SL12
Nombre del término	XAML
Nombre del término en inglés	eXtensible Application Markup Language
Descripción	Lenguaje Extensible de Formato para Aplicaciones en español) es el lenguaje de formato para la interfaz de usuario para la Base de Presentación de Windows (WPF por sus siglas en inglés) y Silverlight(wpf/e), el cual es uno de los "pilares" de la interfaz de programación de aplicaciones .NET en su versión 3.0 (conocida con anterioridad con el nombre clave WinFX). XAML es un lenguaje declarativo basado en XML, optimizado para describir gráficamente interfaces de usuarios visuales ricas desde el punto de vista gráfico, tales como las creadas por medio de Adobe Flash. XUL y UIML son otros ejemplos de lenguajes de interfaz basados en XML. SVG es otro lenguaje de este tipo propuesto por la organización W3C, el cual soporta gráficos, animaciones, audio y video integrados, eventos y comportamiento descrito por medio de escritura y puede ser utilizado como lenguaje de interfaz basado en XML.

XML

Código	SL14
Nombre del término	XML
Nombre del término en inglés	Extensible Markup Language
Descripción	Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos

ASP.NET

Código	ASPNET01
Nombre del término	ASP.NET
Nombre del término en inglés	Active Server Pages for .NET
Descripción	Herramienta de desarrollo Web comercializado por Microsoft. Es usado por programadores para construir sitios Web dinámicos,

	aplicaciones Web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).
--	--

Internet Information Services (IIS)

Código	ASPNET02
Nombre del término	Internet Information Services (IIS)
Nombre del término en inglés	Internet Information Services (IIS)
Descripción	<p>Es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.</p> <p>Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web).</p> <p>El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.</p>

ECMAScript

Código	AJAX01
Nombre del término	ECMAScript
Nombre del término en inglés	ECMAScript
Descripción	<p>Es una especificación de lenguaje de programación publicada por ECMA International. El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation. Actualmente está aceptado como el estándar ISO 16262.</p> <p>ECMAScript define un lenguaje de tipos dinámicos ligeramente inspirado en Java y otros lenguajes del estilo de C. Soporta algunas características de la programación orientada a objetos mediante objetos basados en prototipos y pseudo clases.</p> <p>La mayoría de navegadores de Internet incluyen una implementación del estándar ECMAScript, al igual que un acceso al Document Object Model para manipular páginas Web. JavaScript está implementado en la mayoría de navegadores, y el Internet Explorer de Microsoft usa JScript. El navegador Opera tiene su propio intérprete de ECMAScript con extensiones para soportar algunas características de JavaScript y JScript. Cada navegador tiene extensiones propias al estándar ECMAScript, pero cualquier código que se adecue al estándar debería funcionar en todos ellos.</p> <p>ActionScript, para Macromedia Flash, también está basado en el estándar ECMAScript, con mejoras que permiten a los objetos ser movidos, creados y analizados dinámicamente, mientras la película está en ejecución.</p>

Iframe

Código	AJAX02
Nombre del término	iframe
Nombre del término en inglés	inline frame
Descripción	<p>Es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal. Insertar un iframe entre una sección o bloque es semejante a insertar un elemento object. Esto permite que se pueda insertar un documento HTML dentro de otro, alineado de acuerdo a sus límites.</p> <p>La información insertada en el iframe se especifica por medio del atributo src de este elemento. Sólo los navegadores capaces de mostrar o incluir iframes los aceptarán y reconocerán.</p> <p>El documento incrustado se puede sustituir por otro diferente sin volver a cargar la página que lo rodea: usando el atributo "target" de un enlace HTML o utilizando JavaScript. Ello hace posibles muchas aplicaciones interactivas y por lo tanto es común que se usen iframes en aplicaciones AJAX. La principal alternativa al uso de iframes en estos casos es la edición del árbol DOM del documento. Con frecuencia, se usan iframes invisibles para comunicarse en forma asíncrona con el servidor, como alternativa a usar XMLHttpRequest.</p> <p>Mientras que los frames tradicionales se usan típicamente para hacer subdivisiones lógicas del contenido de un sitio Web, los iframes se usan comúnmente para insertar contenido (como publicidad) de otro sitio Web en la página actual.</p>

JavaScript

Código	AJAX03
Nombre del término	JavaScript
Nombre del término en inglés	JavaScript
Descripción	<p>Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.</p>

AJAX

Código	AJAX04
Nombre del término	AJAX
Nombre del término en inglés	Asynchronous JavaScript And XML
Descripción	<p>Es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.</p>

UML

Código	UML01
Nombre del término	XML
Nombre del término en inglés	Extensible Markup Language
Descripción	Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos

Artefacto

Código	RUP01
Nombre del término	Artefacto
Nombre del término en inglés	
Descripción	Es un elemento de RUP el cual puede ser un documento o un modelo

CamelCase

Código	RUP02
Nombre del término	CamelCase
Nombre del término en inglés	CamelCase
Descripción	Es la práctica de escribir frases o palabras compuestas eliminando los espacios y poniendo en mayúscula la primera letra de cada palabra. El nombre viene del parecido de estas mayúsculas, entre las demás letras, con las jorobas de los camellos.

Casos de uso

Código	RUP03
Nombre del término	Casos de uso
Nombre del término en inglés	Casos de uso
Descripción	Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización software.

Herramientas CASE

Código	RUP04
Nombre del término	Herramientas CASE
Nombre del término en inglés	Herramientas CASE - Computer Aided Software Engineering
Descripción	Las Herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

IEEE

Código	RUP05
Nombre del término	IEEE
Nombre del término en inglés	Institute of Electrical and Electronics Engineers
Descripción	El Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación e ingenieros en telecomunicación.

LowerCamelCase

Código	RUP06
Nombre del término	lowerCamelCase
Nombre del término en inglés	lowerCamelCase
Descripción	Notación empleada para definir nombre mediante el uso de la primera letra de la primera palabra está escrita en minúscula y las primeras letras de las demás palabras están escritas en mayúsculas. Ej: ejemploDeLowerCamelCase

Rational Unified Process

Código	RUP07
Nombre del término	Rational Unified Process
Nombre del término en inglés	Rational Unified Process
Descripción	Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos

Stakeholder

Código	RUP08
Nombre del término	Stakeholder
Nombre del término en inglés	Stakeholder
Descripción	<p>Término fue utilizado para referirse a quienes pueden afectar o son afectados por las actividades de una empresa.</p> <p>Estos grupos o individuos son los públicos interesados ("stakeholders"), deben ser considerados como un elemento esencial en la planeación estratégica de negocios.</p> <p>Ejemplos de Stakeholders</p> <ul style="list-style-type: none"> • Accionistas • Asociaciones empresariales, industriales o profesionales • Clientes • Competidores • Comunidades donde la empresa tiene operaciones: asociaciones vecinales

	<ul style="list-style-type: none"> • Dueños • Empleados • Gobierno nacional • Gobiernos locales • Gobiernos provinciales • Inversores • ONGs • Proveedores/vendedores a la empresa • Sindicatos <p>Todos estos stakeholders sacan beneficios o sufren daños como resultado de las acciones de la misma empresa. De Wikipedia, la enciclopedia libre</p>
--	--

UpperCamelCase

Código	RUP09
Nombre del término	UpperCamelCase
Nombre del término en inglés	UpperCamelCase
Descripción	Notación empleada para especificar nombres usando la primera letra de todas escrita en mayúscula