



UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA
CARRERA DE INGENIERÍA DE SISTEMAS



Evaluación de Calidad Salesforce Módulo de Productos y Listas de Precios

Modalidad: Diplomado de Doble Titulación

Elaborado por: Carlos Eduardo Caballero Burgoa

Cochabamba - Bolivia

Índice general

1. HTSM	1
1.1. Entorno de proyecto	1
1.1.1. Misión	1
1.1.2. Fuentes de Información	2
1.1.3. Equipamiento	2
1.1.4. Cronograma	4
1.1.5. Elementos a evaluar	6
1.1.6. Documentos entregables	6
1.2. Elementos del producto	6
1.2.1. Productos	7
1.2.2. Listas de Precios	7
1.2.3. Controles de Vista de Lista	7
1.3. Técnicas de prueba	7
1.3.1. Pruebas de aceptación	7
1.3.2. Pruebas funcionales	7
1.3.3. Pruebas de dominio	13
1.3.4. Pruebas negativas	13
1.3.5. Pruebas de compatibilidad	13
1.3.6. Pruebas de localización	13
1.4. Criterios de calidad	14
Bibliografía	17

Índice de figuras

1.1. Participación de mercado de los navegadores hasta Noviembre del 2018.	3
1.2. Funciones que componen el módulo de gestión de productos.	8
1.3. Interfaz gráfica para el módulo de productos.	9
1.4. Funciones que componen el módulo de gestión de listas de precios.	10
1.5. Interfaz gráfica para el módulo de listas de precios.	11
1.6. Funciones que componen el módulo de gestión de listas de precios.	12

Índice de cuadros

1.1. Ediciones actualmente disponibles de Sales Cloud.	3
1.2. Ediciones actualmente disponibles de Sales Cloud.	4
1.3. Herramientas auxiliares de apoyo para el proyecto.	4
1.4. Cronograma de actividades del proyecto.	5

1

HTSM

HTSM (Heuristic Test Strategy Model) fue creado por James Bach en 1996 para que lo usen los evaluadores profesionales como una colección estructurada de recordatorios de qué pensar cuando están creando pruebas, divide el pensamiento de la creación de pruebas en diferentes ejes de análisis, que, cuando se unen, permiten al evaluador crear una estrategia de prueba holística[2].

En este capítulo desglosaremos, y describiremos todos sus componentes, de tal forma que este construya un marco sólido para la planificación, y ejecución de las pruebas de calidad del producto.

1.1. Entorno de proyecto

En esta sección se describirán aquellos factores del contexto que incluyen recursos necesarios, restricciones, y cualquier otro elemento del proyecto que debe ser tomado en cuenta para la evaluación.

1.1.1. Misión

La misión del proyecto es:

Evaluar las funcionalidades provistas por *Salesforce* que componen el módulo de gestión de productos y listas de precios, con el diseño y ejecución de múltiples técnicas de evaluación, para que de esta manera se pueda garantizar la calidad del producto para los clientes.

1.1.2. Fuentes de Información

Para evaluar los componentes antes citados, se encontraron las siguientes fuentes de información respecto al producto:

Centro de Ayuda *Salesforce* ofrece un amplio conjunto de documentación, información general, preguntas frecuentes, y contacto con el servicio de asistencia técnica desde su sitio de ayuda (<https://help.salesforce.com/>).

Estos recursos serán útiles para conocer las reclamaciones de los usuarios, las características críticas del producto, y las estrategias del fabricante hacia sus clientes.

Centro de Desarrollo *Salesforce* también posee un sitio web específicamente para compartir recursos de desarrollo sobre la plataforma (<https://developer.salesforce.com/>).

Este sitio se podrá aprovechar para consultar las referencias a las API del servicio, conocer acerca de los componentes y como pueden aprovecharse desde la perspectiva del desarrollador.

Recursos para administradores Sitio web enfocado a ofrecer experiencias, vídeos, herramientas, y un sin fin de recursos orientados a usuarios con el rol de administración de recursos sobre la plataforma (<https://admin.salesforce.com/resources>).

Comunidad Trailblazer Sitio web enfocado a conectar a miembros de la comunidad *Salesforce*, para compartir experiencias, aprender, y proveer de nuevas ideas sobre la utilización del servicio (<https://success.salesforce.com/>).

1.1.3. Equipamiento

En esta sección describiremos la recursos necesarios para la ejecución del proyecto.

Software

Salesforce cuenta con múltiples ediciones que comparten una apariencia, pero varían según la funcionalidad y los precios. Algunos clientes comienzan con una edición básica y actualizan a una edición más rica en características a medida que evolucionan los requisitos empresariales.

En el cuadro 1.1 se describen las ediciones disponibles¹.

Una característica actual de *Salesforce* es que cuenta con dos interfaces web diferentes, la antigua conocida como: *Salesforce Classic*, y la nueva incluida desde 2015 denominada: *Salesforce Lightning*; que tiene como objetivo principal la unificación del *Look and Feel* a través de todo el servicio sea cual sea el dispositivo que el cliente utilice[7].

Se evaluarán las funcionalidades de los módulos sobre el navegador cuya participación en el mercado es la mayor, en este caso: *Google Chrome* como puede verse en la figura 1.1.

¹Información extraída y disponible en: https://help.salesforce.com/articleView?id=overview_edition.htm

Edición	Descripción
<i>Essentials</i>	Herramienta para la elaboración de los mapas mentales.
<i>Professional</i>	Herramienta para la generación de pruebas combinatorias.
<i>Enterprise</i>	Herramienta para la solicitud de peticiones HTTP para el uso de API.
<i>Unlimited</i>	Herramienta basada en web que facilita el proceso de gestión de calidad de software.
<i>Developer</i>	Servicio de almacenamiento y versionado de código fuente y documentación.

Cuadro 1.1: Ediciones actualmente disponibles de Sales Cloud.

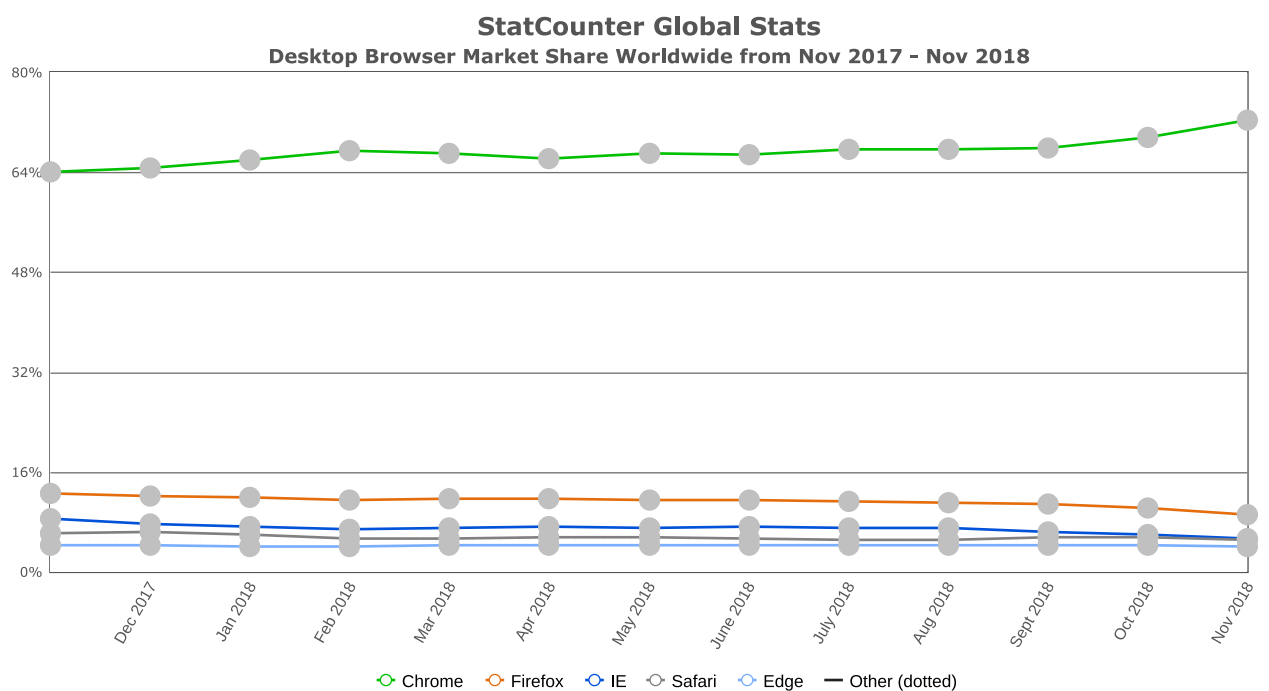


Figura 1.1: Participación de mercado de los navegadores hasta Noviembre del 2018.

	Microsoft Internet Explorer	Microsoft Edge	Google Chrome	Mozilla Firefox	Apple Safari
Lightning Experience	IE11 (EOL Diciembre 31, 2020)	Ultima versión	Ultima versión	Ultima versión	11.x+
Lightning Communities	IE11 (EOL Diciembre 31, 2020)	Ultima versión	Ultima versión	Ultima versión	11.x+
¿Consideraciones especiales de configuración?	No	No	No	No	No
Limitaciones conocidas	Sí	Si	No	Si	Si

Cuadro 1.2: Ediciones actualmente disponibles de Sales Cloud.

Herramienta	Descripción
<i>XMind</i>	Herramienta para la elaboración de los mapas mentales.
<i>GitHub.com</i>	Servicio de almacenamiento y versionado de código fuente y documentación.
<i>Testlink</i>	Herramienta basada en web que facilita el proceso de gestión de calidad de software.
<i>LambdaTest</i>	Herramienta basada en web para realizar la evaluación de compatibilidad en navegadores.
<i>Pairwise</i>	Herramienta para la generación de pruebas combinatorias.
<i>Postman</i>	Herramienta para la solicitud de peticiones HTTP para el uso de API.

Cuadro 1.3: Herramientas auxiliares de apoyo para el proyecto.

Adicionalmente se utilizaran algunos navegadores más en pruebas muy focalizadas de compatibilidad. A partir de la información disponible en la pagina de soporte provista por el fabricante. Como puede verse en el cuadro 1.2².

Herramientas

En diferentes etapas del proyecto se utilizaran múltiples herramientas que colaboren con tareas específicas del proceso de evaluación, ellas detalladas en el cuadro 1.3.

1.1.4. Cronograma

Véase el cuadro 1.4 en la página 5.

²Información extraída y disponible en: https://help.salesforce.com/articleView?id=getstart.browsers_sfx.htm

Objetivo General	Objetivos Específicos	Actividades	Resultados
Evaluar el grado de eficacia y eficiencia de los módulos de productos y listas de precios para garantizar la calidad del servicio hacia el cliente.	Evaluar las funciones del módulo de gestión de productos para garantizar sus atributos de calidad.	Recolectar información, explorar y analizar el módulo de productos.	Batería de pruebas de evaluación del modulo de productos.
		Diseñar los tipos de evaluación requeridos para el módulo de productos.	
		Construir la batería de pruebas necesarias para el módulo de productos.	
	Evaluar las funciones del módulo de gestión de listas de precios para garantizar sus atributos de calidad.	Recolectar información, explorar y analizar el módulo de listas de precios.	Batería de pruebas de evaluación del modulo de listas de precios.
		Diseñar los tipos de evaluación requeridos para el módulo de listas de precios.	
		Construir la batería de pruebas necesarias para el módulo de listas de precios.	
	Evaluar las características de localización, compatibilidad, y calidad del soporte ofrecido.	Explorar, analizar, diseñar y construir una batería de pruebas de localización para los módulos a evaluar.	Batería de pruebas de localización, compatibilidad, y soportabilidad.
		Explorar, analizar, diseñar y construir una batería de pruebas de compatibilidad para los módulos a evaluar.	
		Explorar, analizar, diseñar y construir una batería de pruebas de soportabilidad para los módulos a evaluar.	
	Evaluar el comportamiento del API disponible del servicio, relativo a productos y listas de precios.	Recolectar información, explorar, y analizar las peticiones disponibles en el servicio que comprenden el módulo a evaluar.	Batería de pruebas automatizadas sobre el API disponible.
		Diseñar los tipos de evaluación requeridos para el API disponible.	
		Construir y automatizar la batería de pruebas necesarias para la evaluación del API.	
	Ejecutar las baterías de pruebas construidas para garantizar la calidad general de los componentes evaluados.	Realizar la ejecución de las pruebas manuales.	Informe de ejecución de pruebas.
		Realizar la ejecución de las pruebas automatizadas.	
		Analizar y construir estadísticas finales de la evaluación.	

Cuadro 1.4: Cronograma de actividades del proyecto.

1.1.5. Elementos a evaluar

Los elementos que se evaluarán los siguientes:

- Productos.
- Precios de productos.
- Listas de precios.

La evaluación se realizará a la plataforma, mediante la utilización del navegador web Google Chrome en su versión 68.0, con excepción de las pruebas automatizadas que serán ejecutadas mediante scripts de automatización, y las pruebas de compatibilidad que requerirán muchos navegadores para su evaluación.

1.1.6. Documentos entregables

A continuación se describen los documentos resultantes de la evaluación:

HTSM Documento de describe todos los elementos necesarios par la creación de pruebas.

Batería de casos de prueba Conjunto de pruebas a realizar para garantizar la calidad del producto.

Batería de reportes de error Conjunto de reportes de error, generados de cualquier problema encontrado en el proceso de ejecución de las pruebas.

Matriz de trazabilidad Documento que describe la Relación de cobertura de las funcionalidades a evaluar con los casos de prueba diseñados.

Scripts de automatización Código fuente de los programas que se encargan de la automatización de las pruebas sobre el API del servicio.

Resultados de ejecución Informe técnico que condensa los resultados finales de la ejecución de las pruebas.

1.2. Elementos del producto

Dentro del alcance de la evaluación se encuentran los componentes de productos y listas de precios, las funcionalidades que comprenden estos se detallan en esta sección.

Se considero la interfaz *Lightning Experience*, como único objetivo de la evaluación. La versión *Lightning Experience* esta disponible para las siguientes ediciones del producto: *Essentials*, *Group*, *Professional*, *Enterprise*, *Performance*, *Unlimited*, y *Developer*.

1.2.1. Productos

En la figura 1.2 pueden verse las funcionalidades clasificadas desde la perspectiva de la interfaz de usuario, en ambos componentes se tienen la parte de los controles de vista de lista similares, por lo que se prefirió describir esas funciones a una estructura posterior.

Esta clasificación de las funciones se ubican en una vista de interfaz como puede verse en la figura 1.3, vista principal para la gestión de productos en el servicio.

1.2.2. Listas de Precios

En la figura 1.4 pueden verse las funcionalidades clasificadas desde la perspectiva de la interfaz de usuario, también se omitió la sección de los controles de vista de lista.

Esta clasificación de las funciones se ubican en una vista de interfaz como puede verse en la figura 1.5, vista principal para la gestión de listas de precios en el servicio.

1.2.3. Controles de Vista de Lista

En la figura 1.6 pueden verse las funciones omitidas en los diagramas anteriores relativas a los controles de vista, ambas equivalentes entre si.

1.3. Técnicas de prueba

Habiéndose definido los elementos del sistema, y los criterios bajo los que estos deben ser evaluados. Ahora se pasará a describir las técnicas de prueba que se utilizarán para cada elemento y criterio de calidad.

1.3.1. Pruebas de aceptación

La prueba de aceptación es una prueba formal que se realiza para determinar si un sistema satisface sus criterios de aceptación: los criterios que debe cumplir el sistema para que el cliente los acepte. Ayuda al cliente a determinar si acepta o no el sistema[10].

1.3.2. Pruebas funcionales

El software o sistema bajo prueba se ve como una «caja negra». La selección de casos de prueba para pruebas funcionales se basa en el requisito o especificación de diseño de la entidad de software bajo prueba. Ejemplos de resultados esperados, algunas veces se llaman oráculos de prueba, incluyen requisitos/especificaciones de diseño, valores calculados a mano y resultados simulados. Las pruebas funcionales hacen hincapié en el comportamiento externo de la entidad de software[6].

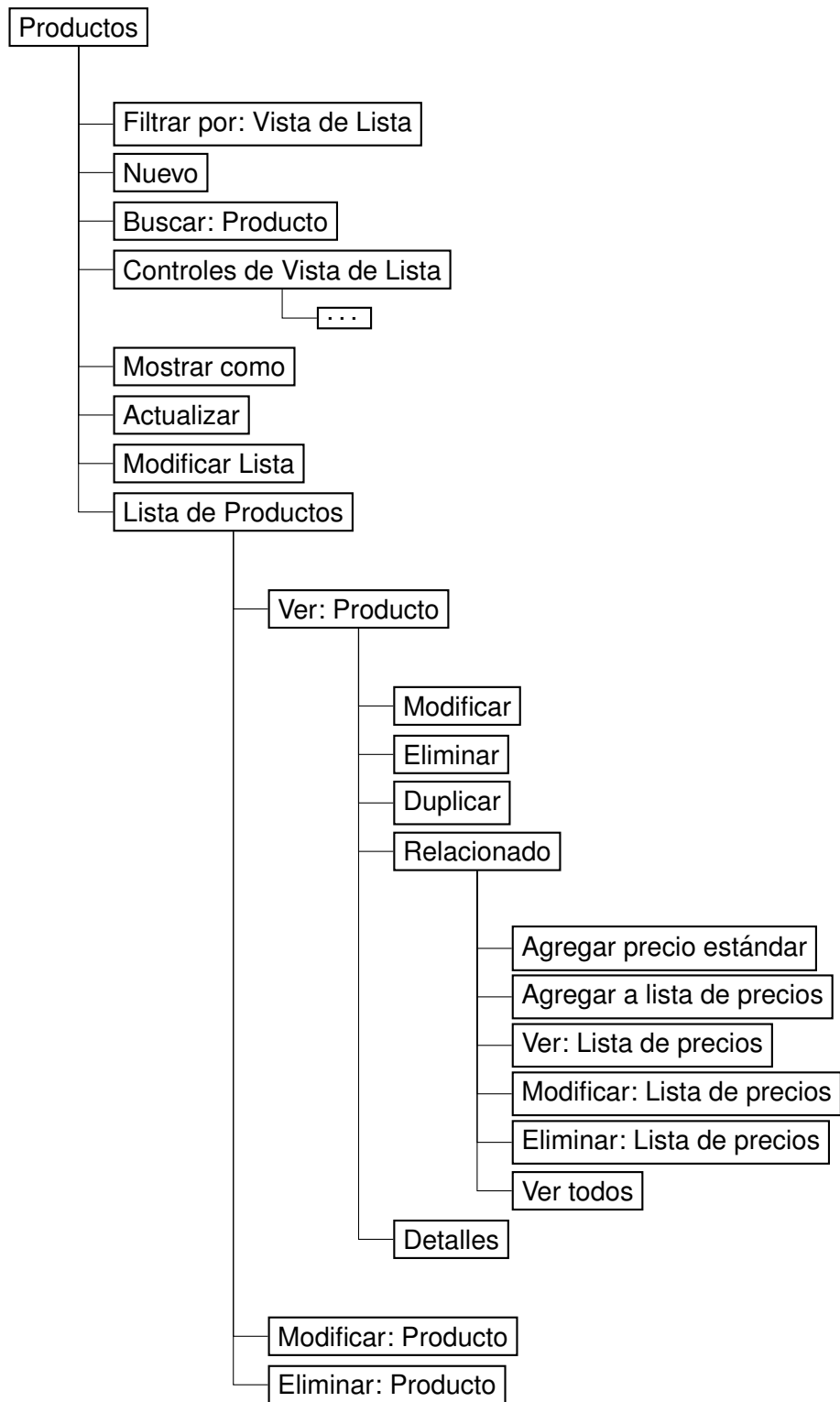


Figura 1.2: Funciones que componen el módulo de gestión de productos.

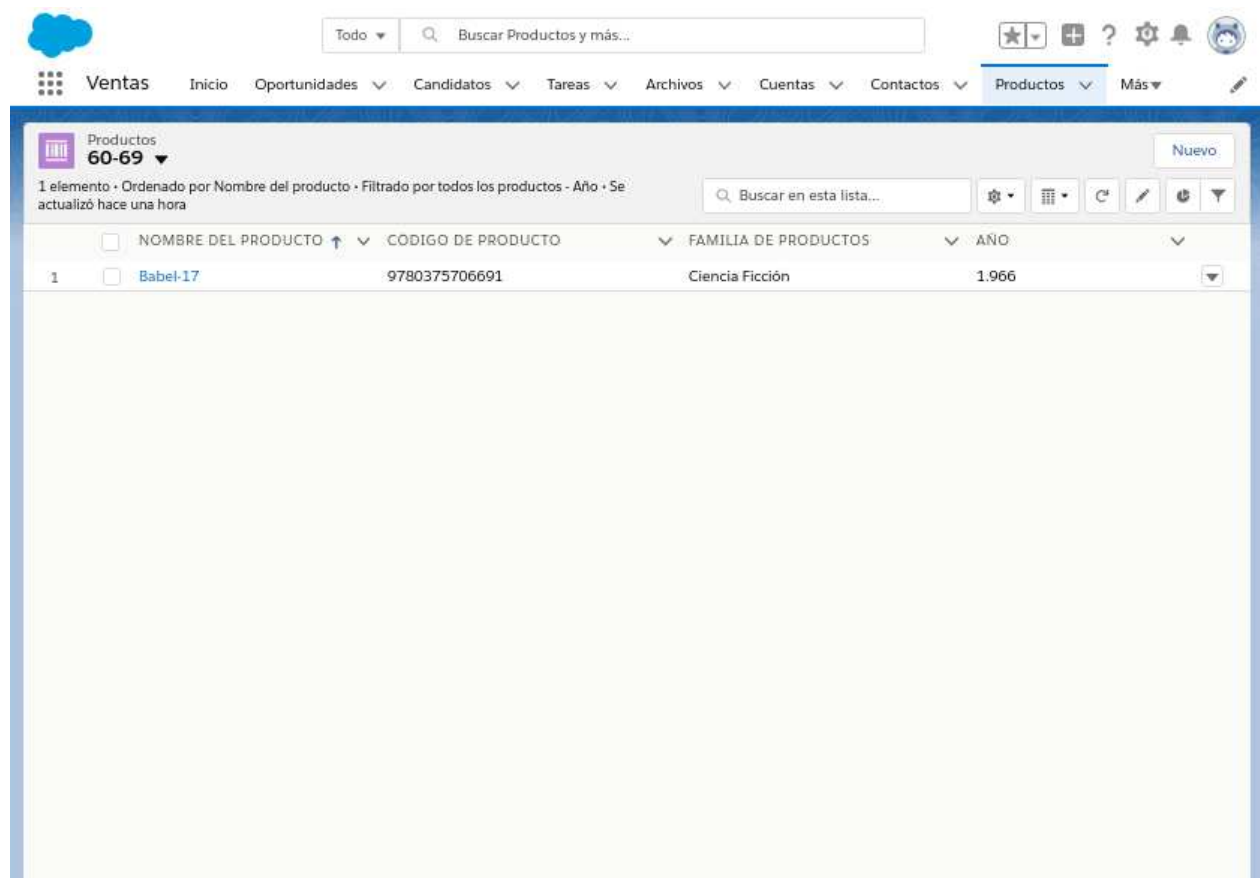


Figura 1.3: Interfaz gráfica para el módulo de productos.

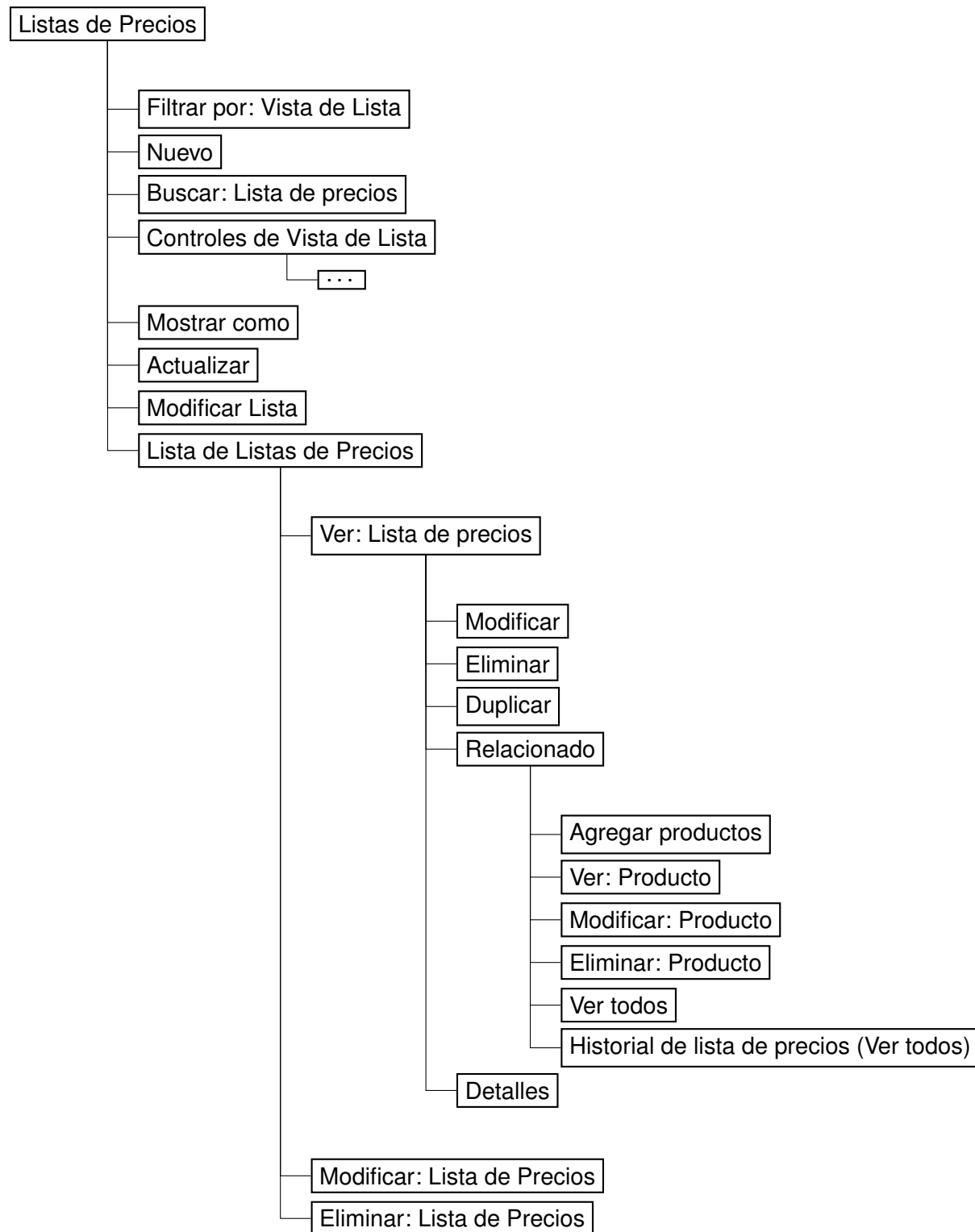


Figura 1.4: Funciones que componen el módulo de gestión de listas de precios.

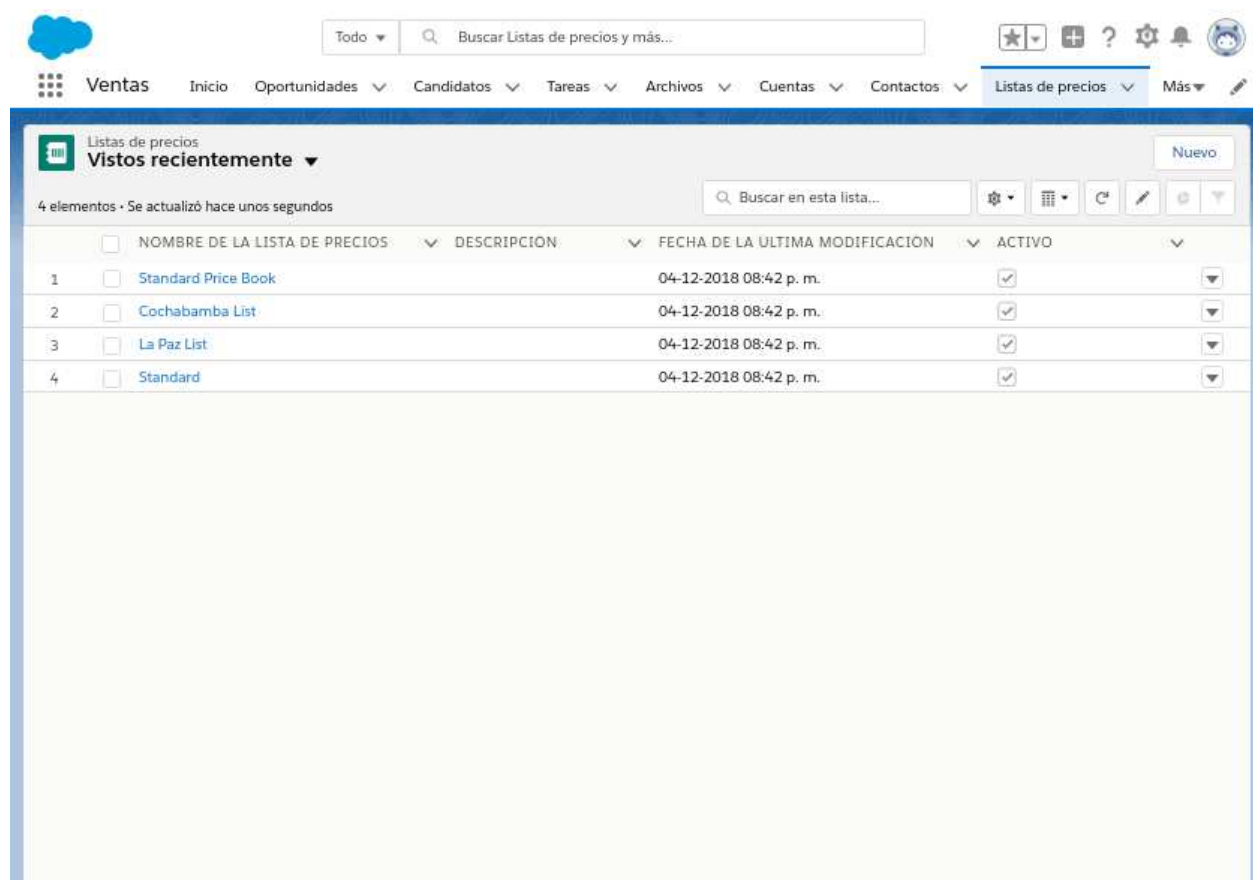


Figura 1.5: Interfaz gráfica para el módulo de listas de precios.

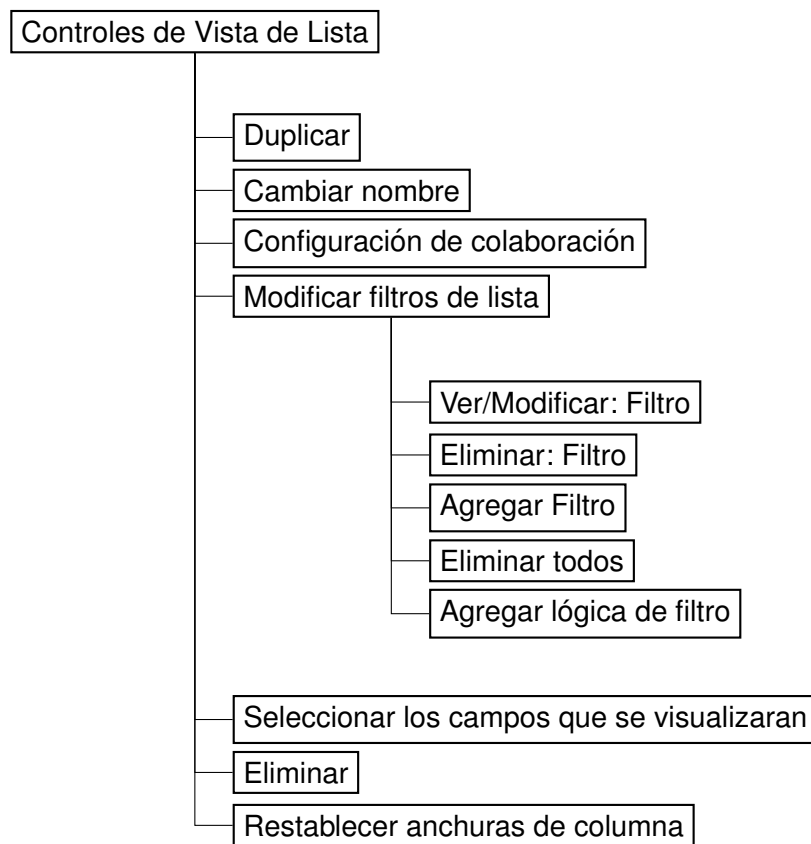


Figura 1.6: Funciones que componen el módulo de gestión de listas de precios.

1.3.3. Pruebas de dominio

La prueba de dominio es una estrategia de muestreo estratificada para elegir algunos casos de prueba de la infinidad de casos de prueba candidatos. La estrategia tiene varios nombres, como la partición de equivalencia, el análisis de límites y la partición de categorías.

La prueba de dominio es probablemente la más ampliamente descrita y una de las técnicas de prueba de software más ampliamente practicadas. Algunos autores restringen su consideración del alcance de esta técnica a variables de entrada linealizables a funciones matemáticas. Una variable linealizable es aquella cuyos valores se pueden asignar a una recta numérica. El análisis es más sencillo y más obvio en estos casos[5].

1.3.4. Pruebas negativas

La prueba negativa, comúnmente conocida como *prueba de ruta de error* o *prueba de falla*, generalmente se realiza para garantizar la estabilidad de la aplicación.

La prueba negativa es el proceso de aplicar tanta creatividad como sea posible y validar la aplicación contra datos no válidos. Esto significa que su propósito es verificar si los errores se muestran al usuario donde se supone que debe hacerlo o si se está manejando un valor incorrecto con mayor gracia.

La fiabilidad funcional de la aplicación o el software solo se puede cuantificar con escenarios negativos diseñados de manera efectiva. Las pruebas negativas no solo apuntan a detectar fallas potenciales que podrían causar un impacto grave en el consumo del servicio, sino que también pueden ser fundamentales para determinar las condiciones bajo las cuales la aplicación puede fallar. Finalmente, garantiza que haya suficiente validación de errores presente en el software[9].

1.3.5. Pruebas de compatibilidad

La prueba de compatibilidad es una prueba no funcional realizada en la aplicación para evaluar la compatibilidad de la aplicación en diferentes entornos. En este caso sobre diferentes navegadores.

1.3.6. Pruebas de localización

El propósito de las pruebas de localización es asegurar que los errores no se hayan introducido durante el proceso de traducción, que el contenido traducido se muestre correctamente y que el producto localizado funcione como se espera para el mercado objetivo. Probar sitios web para garantizar la adaptabilidad a un lugar y una cultura en particular también es crucial para sugerir su superioridad en el mercado objetivo[11].

Aquí hay algunas áreas que examinamos durante esta prueba:

- Todos los recursos de localización están traducidos correctamente.

- La compilación generada incluye todos los archivos necesarios.
- La funcionalidad en la versión localizada es consistente con el producto de origen.
- La pantalla localizada tiene el mismo número y tipo de elementos que el del producto de origen.
- Todos los caracteres específicos del entorno local aparecen correctamente.
- No hay palabras que pasen sobre los botones o se corten en la página.

1.4. Criterios de calidad

Se denomina criterio de calidad a cualquier requerimiento que define lo que el producto debe ser.

Por lo general, los criterios de calidad parten de la combinación de las necesidades reales y de las demandas de los clientes, con el conocimiento de las ofertas y productos de organizaciones de la competencia y las posibilidades que el fabricante posee para satisfacer esas necesidades y expectativas o para procurar en la medida de lo posible y/o aconsejable[4].

Se definieron los siguientes como criterios de calidad fundamentales para el éxito del producto[3].

Usabilidad La usabilidad se refiere a la facilidad de operación del producto por parte de los usuarios, y se relaciona con el esfuerzo necesario para ser utilizado, y en la evaluación individual de tal uso, por parte de un conjunto especificado o implícito de usuarios.

Entre algunos de los criterios que determinan este factor, están:

- **Entendimiento** que mide el esfuerzo del usuario en reconocer el concepto lógico del software y su aplicabilidad.
- **Aprendizaje** que mide el esfuerzo del usuario en aprender acerca del producto.
- **Operabilidad** que mide el esfuerzo del usuario en operar y controlar el sistema.

Compatibilidad La compatibilidad del navegador determina el comportamiento del servicio en diferentes plataformas de navegación.

Dado que cada navegador tiene su propia manera de mostrar y gestionar los contenidos de una página web. Por lo tanto, las páginas web deben diseñarse de tal manera que puedan ser compatibles con cada uno de los navegadores de uso común. Actualmente hay casi cien tipos diferentes de navegadores disponibles, lo que dificulta que los diseñadores / webmasters desarrollen sitios web con un comportamiento similar en múltiples plataformas. El estricto cumplimiento de las pautas de diseño puede cumplir con estos criterios hasta cierto nivel.

Soportabilidad La soportabilidad es la capacidad del sistema para proporcionar información útil para identificar y resolver problemas.

El costo de mantener el atributo de compatibilidad es alto y el resultado solo es visible a gran escala. Sin embargo, con el crecimiento del equipo y el producto, este atributo se convierte en una de las claves[1].

Localizabilidad La localizabilidad es un proceso intermedio para verificar que una aplicación globalizada está lista para la localización. En una situación ideal, esta es solo una fase de garantía de calidad. Si se diseñó y desarrolló una aplicación con miras a la localización, esta fase consistirá principalmente en pruebas de localizabilidad. De lo contrario, es durante esta fase que se descubrirán y corregirán los errores en el código fuente que impiden la localización.

La localizabilidad ayuda a garantizar que la localización no introduzca ningún defecto funcional en la aplicación[8].

Bibliografía

- [1] Ashanin, Nikolay.
Quality attributes in Software Architecture. Part I
Extraído el 12 de Diciembre del 2018, de
<https://hackernoon.com/quality-attributes-in-software-architecture-3844ea482732>
- [2] Bach, James.
Heuristic Test Strategy Model.
Extraído el 01 de Diciembre del 2018, de
<https://www.satisfice.com/tools/htsm.pdf>
- [3] Fillottrani, Pablo R..
Calidad en el Desarrollo de Software
Extraído el 12 de Diciembre del 2018, de
<http://www.cs.uns.edu.ar/prf/teaching/SQ07/clase6.pdf>
- [4] Haaz Díaz, Alberto.
Criterios, Indicadores y Estándares de Calidad
Extraído el 12 de Diciembre del 2018, de
<https://haaz-calidad.blogspot.com/2010/05/criterios-indicadores-y-estandares-de.html>
- [5] Kaner, Cem.
Teaching Domain Testing: A Status Report
Extraído el 12 de Diciembre del 2018, de
<http://www.testineducation.org/a/tdtsr.pdf>
- [6] Luo, Lu.
Software Testing Techniques
Extraído el 12 de Diciembre del 2018, de
<https://www.cs.cmu.edu/~luluo/Courses/17939Report.pdf>
- [7] McCarthy, Ben.
A Brief History Of Salesforce.com.
Extraído el 12 de Diciembre del 2018, de
<https://www.salesforceben.com/brief-history-salesforce-com/>
- [8] Moura, Jorge.
Globalization, localizability, locatization in software testing - part 1
Extraído el 12 de Diciembre del 2018, de
<https://ojmoura.wordpress.com/2008/12/26/globalization-localizability-locatization-in-software-testing-part-1/>

- [9] Nadig, Sneha.
What is Negative Testing and How to Write Negative Test Cases?
Extraído el 12 de Diciembre del 2018, de
<https://www.softwaretestinghelp.com/what-is-negative-testing/>
- [10] Naik, Kshirasagar.
Software Testing and Quality Assurance
Extraído el 12 de Diciembre del 2018, de
<http://ebooks.bharathuniv.ac.in/gdlc1/gdlc1/Software>
- [11] Sampair, Gretchen.
3 critical localization testing methods for software and websites
Extraído el 13 de Diciembre del 2018, de
<https://blog.amplexor.com/globalcontent/en/localization-testing-methods-software-websites>