



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS E INFORMÁTICA**

AUTOR: BERMEO RODRÍGUEZ, LUIS ISRAEL

**TEMA: ANÁLISIS COMPARATIVO DE FRAMEWORKS JAVASCRIPT:
JQUERY Y MOOTOOLS, PARA LA IMPLEMENTACIÓN DE
APLICACIONES WEB EN LA EMPRESA SOFYA. APLICACIÓN A UN
CASO DE ESTUDIO**

DIRECTOR: ING. CAMPAÑA MAURICIO

CODIRECTOR: ING. MELO LORENA

SANGOLQUÍ, AGOSTO 2014

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICADO

Ing. Mauricio Campaña

Ing. Lorena Melo

CERTIFICAN

Que el trabajo titulado “ANÁLISIS COMPARATIVO DE FRAMEWORKS JAVASCRIPT: JQUERY Y MOOTOOLS, PARA LA IMPLEMENTACIÓN DE APLICACIONES WEB EN LA EMPRESA SOFYA. APLICACIÓN A UN CASO DE ESTUDIO.”, realizado por el Sr. Luis Israel Bermeo Rodríguez, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas – ESPE.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (PDF). Se autoriza al Sr. Luis Israel Bermeo Rodríguez, que el material se entregue al Ing. Mauricio Campaña, en su calidad de Director de la Carrera.

Sangolquí, 9 de Agosto de 2014

Ing. Mauricio Campaña
DIRECTOR

Ing. Lorena Melo
CODIRECTORA

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

DECLARACIÓN DE RESPONSABILIDAD

Yo, Luis Israel Bermeo Rodríguez

DECLARO QUE:

El proyecto de grado denominado “ANÁLISIS COMPARATIVO DE FRAMEWORKS JAVASCRIPT: JQUERY Y MOOTOOLS, PARA LA IMPLEMENTACIÓN DE APLICACIONES WEB EN LA EMPRESA SOFYA. APLICACIÓN A UN CASO DE ESTUDIO.”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mí autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 9 de Agosto de 2014

Luis Bermeo R.

CC: 1723645238

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

AUTORIZACIÓN DE PUBLICACIÓN

Yo, Luis Israel Bermeo Rodríguez, autorizo a la Universidad de las Fuerzas Armadas – ESPE la publicación, en la biblioteca virtual de la institución, del trabajo “ANÁLISIS COMPARATIVO DE FRAMEWORKS JAVASCRIPT: JQUERY Y MOOTOOLS, PARA LA IMPLEMENTACIÓN DE APLICACIONES WEB EN LA EMPRESA SOFYA. APLICACIÓN A UN CASO DE ESTUDIO.”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Consecuentemente este trabajo es de mí autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 9 de Agosto de 2014

Luis Bermeo R.
CC: 1723645238

DEDICATORIA

Dedico este proyecto a mis padres Salvador e Hilda, ya que han sido el apoyo incondicional para lograr mis objetivos tanto académicos como personales. La confianza y el amor que me han brindado en el transcurso de mi vida han sido crucial para poder culminar este proyecto. También dedico este logro a mis hermanos, hermanas y amigos que de alguna u otra forma a través de opiniones y consejos han aportado con un granito de arena al esfuerzo realizado.

AGRADECIMIENTO

Agradezco a Dios por brindarme salud y sabiduría para elegir y alcanzar los objetivos correctos propuestos hasta el momento.

A mis padres Salvador e Hilda que no han descansado por ver a todos sus hijos crecer sanos, inteligentes, perseverantes y comprometidos con la vida.

A mis maestros que día a día en el transcurso de mi carrera académica han aportado conocimientos, experiencias y buenas prácticas.

Agradezco a mis hermanos y hermanas que siempre han estado a mi lado ayudándome en todo el proceso de formación personal y profesional.

También agradezco a mis amigos y compañeros por el apoyo moral y académico que me han brindado durante estos cinco años.

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS	VII
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XI
CAPÍTULO 1: ANÁLISIS COMPARATIVO DE FRAMEWORKS JAVASCRIPT: JQUERY Y MOOTOOLS, PARA LA IMPLEMENTACIÓN DE APLICACIONES WEB EN LA EMPRESA SOFYA. APLICACIÓN A UN CASO DE ESTUDIO.	1
1.1. ANTECEDENTES	1
1.2. PLANTEAMIENTO DEL PROBLEMA	1
1.3. JUSTIFICACIÓN	2
1.4. OBJETIVOS	3
1.4.1. GENERAL	3
1.4.2. ESPECÍFICOS	3
1.5. ALCANCE	3
CAPÍTULO 2: MARCO TEÓRICO DE REFERENCIA	5
2.1. USO DE FRAMEWORKS JAVASCRIPT	5
2.1.1. INTRODUCCIÓN A LOS FRAMEWORKS	5
2.1.2. RAZONES PARA UTILIZAR UN FRAMEWORK JAVASCRIPT	5
2.2. FRAMEWORK JQUERY	6
2.2.1. INTRODUCCIÓN	6
2.2.2. CARACTERÍSTICAS	7
2.2.3. POO. BASADA EN PROTOTIPOS	7
2.2.4. AJAX	9
2.2.5. MANIPULACIÓN DEL DOM	10
2.2.6. VALIDACIÓN DE FORMULARIOS	10
2.2.7. EFECTOS VISUALES	11
2.2.8. MANEJO DE EVENTOS	12
2.2.9. JQUERY PARA MÓVILES	12
2.2.10. PRUEBAS UNITARIAS	14
2.2.11. INSTALACIÓN DE JQUERY	14
2.2.12. INFORMACIÓN TÉCNICA	15
2.2.13. VERSIONES DE JQUERY	16
2.2.14. COMUNIDAD	16
2.3. FRAMEWORK MOOTOOLS	17
2.3.1. INTRODUCCIÓN	17
2.3.2. CARACTERÍSTICAS	17
2.3.3. POO. BASADA EN CLASES	18
2.3.4. AJAX	19
2.3.5. MANIPULACIÓN DEL DOM	21
2.3.6. VALIDACIÓN DE FORMULARIOS	22
2.3.7. EFECTOS VISUALES	23
2.3.8. MANEJO DE EVENTOS	24
2.3.9. MOOTOOLS PARA MÓVILES	25
2.3.10. INSTALACIÓN DE MOOTOOLS	26
2.3.11. INFORMACIÓN TÉCNICA	26
2.3.12. VERSIONES DE MOOTOOLS	27

2.3.13. COMUNIDAD	27
2.4. NORMA ISO/IEC 25000.....	27
2.4.1. INTRODUCCIÓN.....	27
2.4.2. DIVISIONES DE LA NORMA ISO-25000	28
2.4.3. FAMILIA ISO/IEC 25010	29
2.4.4. FUNCIONALIDAD	31
2.4.5. RENDIMIENTO	31
2.4.6. COMPATIBILIDAD	32
2.4.7. USABILIDAD	32
2.4.8. FIABILIDAD.....	33
2.4.9. SEGURIDAD	33
2.4.10. MANTENIBILIDAD.....	34
2.4.11. PORTABILIDAD	34
2.5. MODELOS DE CALIDAD.....	35
2.5.1. TIPOS DE MODELOS DE CALIDAD	35
2.5.1.1. MODELOS DE CALIDAD FIJOS	35
2.5.1.2. MODELOS DE CALIDAD A MEDIDA	36
2.5.1.3. MODELOS DE CALIDAD MIXTOS	36
2.5.2. PROPIEDADES DE LOS MODELOS DE CALIDAD	37
2.6. MÉTODO IQMC.....	39
2.6.1. INTRODUCCIÓN AL MÉTODO IQMC	39
2.6.2. ETAPAS O PASOS DEL MODELO IQMC	40
2.6.2.1. PASO 0: ESTUDIO DEL ÁMBITO DEL SOFTWARE.....	40
2.6.2.2. PASO 1: DETERMINACIÓN DE CARACTERÍSTICAS DE CALIDAD	40
2.6.2.3. PASO 2: REFINAMIENTO DE LA JERARQUÍA DE SUBCARACTERÍSTICAS	41
2.6.2.4. PASO 3: REFINAMIENTO DE SUBCARACTERÍSTICAS EN ATRIBUTOS	41
2.6.2.5. PASO 4: REFINAMIENTO DE ATRIBUTOS DERIVADOS EN BÁSICOS.....	41
2.6.2.6. PASO 5: ESTABLECIMIENTO DE RELACIONES ENTRE FACTORES DE CALIDAD	41
2.6.2.7. PASO 6: DETERMINACIÓN DE MÉTRICAS PARA LOS ATRIBUTOS	42
2.7. METODOLOGÍA UWE	42
2.7.1. INTRODUCCIÓN.....	42
2.7.2. ETAPAS	42
2.7.2.1. ETAPA 1: ANÁLISIS DE REQUISITOS.....	42
2.7.2.2. ETAPA 2: DISEÑO DE CONTENIDO.....	43
2.7.2.3. ETAPA 3: DISEÑO DE NAVEGACIÓN	43
2.7.2.4. ETAPA 4: DISEÑO DE PRESENTACIÓN	44
2.7.2.5. ETAPA 5: DISEÑO DE PROCESOS	44
CAPÍTULO 3: ESTUDIO COMPARATIVO.....	45
3.1. INTRODUCCIÓN.....	45
3.2. CONSTRUCCIÓN DEL MODELO DE EVALUACIÓN DE CALIDAD.....	45
3.2.1. APLICACIÓN DEL MÉTODO IQMC.....	45
3.2.1.1. PASO 0: ESTUDIO DEL ÁMBITO DEL SOFTWARE.....	45
3.2.1.2. PASO 1: DETERMINACIÓN DE CARACTERÍSTICAS DE CALIDAD	48
3.2.1.3. PASO 2: REFINAMIENTO DE LA JERARQUÍA DE SUBCARACTERÍSTICAS DE CALIDAD	49

3.2.1.4. PASO 3: REFINAMIENTO DE SUBCARACTERÍSTICAS EN ATRIBUTOS	55
3.2.1.5. PASO 4: REFINAMIENTO DE ATRIBUTOS DERIVADOS EN BÁSICOS.....	71
3.2.1.6. PASO 5: ESTABLECIMIENTO DE RELACIONES ENTRE FACTORES DE CALIDAD	74
3.2.1.7. PASO 6: DETERMINACIÓN DE MÉTRICAS PARA LOS ATRIBUTOS	75
3.2.2. MODELO INTEGRADO	77
3.3. APLICACIÓN DEL MODELO	81
3.4. RESULTADO DEL ANÁLISIS COMPARATIVO	83
3.5. INTERPRETACIÓN	85
CAPÍTULO 4: DESARROLLO DE LA APLICACIÓN SPCP (SISTEMA DE COMERCIALIZACIÓN Y PRODUCCIÓN DE CONCRETO)	86
4.1. ANÁLISIS DE REQUISITOS	86
4.1.1. IDENTIFICACIÓN DE LOS ACTORES (USUARIOS):.....	86
4.1.2. CASOS DE USO.....	88
4.1.3. DESCRIPCIÓN DE LOS CASOS DE USO	90
4.2. DISEÑO CONCEPTUAL.....	134
4.3. DISEÑO DE NAVEGACIÓN.....	136
4.4. DISEÑO DE PRESENTACIÓN.	138
4.4.1. PÁGINA GENÉRICA.....	138
4.4.2. INFORMACIÓN DE LA EMPRESA	140
4.4.3. AGENDA DE PEDIDOS	141
4.4.4. INSPECCIÓN	142
4.4.5. AGENDA DE DESPACHOS.....	143
4.4.6. DOCUMENTOS: "ORDEN DE PEDIDO" Y "GUÍA DE REMISIÓN" ..	144
4.5. DISEÑO DE PROCESOS	145
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES	146
5.1. CONCLUSIONES.....	146
5.2. RECOMENDACIONES.....	147
BIBLIOGRAFÍA	149

ÍNDICE DE FIGURAS

Figura 1	Petición Ajax, comparación entre Framework jQuery y JavaScript.....	6
Figura 2	Programación Orientada a Objetos basada en Prototipos.....	8
Figura 3	Programación Orientada a Objetos basada en clases.....	8
Figura 4	Sintaxis MooTools para instanciar una clase.....	18
Figura 5	Ejemplo de uso de Clases en MooTools.....	19
Figura 6	Divisiones de la Norma ISO/IEC 25000.....	28
Figura 7	Características (Internas y Externas) de Calidad del Producto Software.....	30
Figura 8	Relaciones Entre Factores de Calidad.....	75
Figura 9	Resultado del Análisis Comparativo de los Frameworks.....	84
Figura 10	Casos de Uso del Sistema.....	89
Figura 11	Diagrama de Clases del Sistema.....	135
Figura 12	Diseño de Navegación del Sistema.....	137
Figura 13	Página Genérica.....	139
Figura 14	Página: Información Empresa.....	140
Figura 15	Página: Agenda de Pedidos.....	141
Figura 16	Página: Inspección.....	142
Figura 17	Página: Agenda de Órdenes de Pedido.....	143
Figura 18	Páginas: Orden de Pedido y Guía de Remisión.....	144
Figura 19	Modelo de la estructura de procesos.....	145

ÍNDICE DE TABLAS

Tabla 1 Información técnica del Framework jQuery	15
Tabla 2 Versiones del Framework jQuery.....	16
Tabla 3 Información técnica del Framework MooTools	26
Tabla 4 Etapas del Método IQMC.....	40
Tabla 5 Características de Calidad de un Framework JavaScript.....	48
Tabla 6 Subcaracterísticas de Funcionalidad	49
Tabla 7 Subcaracterísticas de Rendimiento	50
Tabla 8 Subcaracterísticas de Compatibilidad.....	50
Tabla 9 Subcaracterísticas de Usabilidad.....	51
Tabla 10 Subcaracterísticas de Fiabilidad	52
Tabla 11 Subcaracterísticas de Seguridad.....	52
Tabla 12 Subcaracterísticas de Mantenibilidad	53
Tabla 13 Subcaracterísticas de Portabilidad	54
Tabla 14 Atributos de Integridad.....	55
Tabla 15 Atributos de Precisión	56
Tabla 16 Atributos de Idoneidad	56
Tabla 17 Atributos de Comportamiento en el Tiempo	57
Tabla 18 Atributos de Rendimiento.....	58
Tabla 19 Atributos de Capacidad de Operación	58
Tabla 20 Atributos de Coexistencia	59
Tabla 21 Atributos de Compatibilidad	59
Tabla 22 Atributos de Inteligibilidad	60
Tabla 23 Atributos de Facilidad de Aprendizaje	60
Tabla 24 Atributos de Operatividad	61
Tabla 25 Atributos de Protección de Errores de Usuario	61
Tabla 26 Atributos de Estética	62
Tabla 27 Atributos de Accesibilidad.....	62
Tabla 28 Atributos de Madurez.....	62
Tabla 29 Atributos de Disponibilidad	63
Tabla 30 Atributos de Tolerancia a Fallos	63
Tabla 31 Atributos de Capacidad de Recuperación	64
Tabla 32 Atributos de Confidencialidad	64
Tabla 33 Atributos de Integridad.....	65
Tabla 34 Atributos de No Repudio.....	65
Tabla 35 Atributos de Responsabilidad	66
Tabla 36 Atributos de Autenticidad	66
Tabla 37 Atributos de Modularidad	66
Tabla 38 Atributos de Reusabilidad	67
Tabla 39 Atributos de Capacidad de ser Analizado.....	67
Tabla 40 Atributos de Capacidad de ser Modificado	68
Tabla 41 Atributos de Capacidad de ser Probado.....	68
Tabla 42 Atributos de Adaptabilidad	68
Tabla 43 Atributos de Facilidad de Instalación	69
Tabla 44 Atributos de Capacidad de ser Reemplazado	70
Tabla 45 Atributos básicos de Efectividad en la creación de efectos y animaciones.....	71
Tabla 46 Atributos Básicos de Efectividad en la Manipulación del DOM	72
Tabla 47 Atributos básicos de Sintaxis	72
Tabla 48 Atributos básicos de Semántica.....	73
Tabla 49 Atributos básicos de Descarga del Framework	73
Tabla 50 Atributos básicos de Referenciar el Framework	74
Tabla 51 Métrica de Cumplimiento	76
Tabla 52 Métrica de Cumplimiento por Rangos.....	76
Tabla 53 Modelo de Evaluación de Calidad	77
Tabla 54 Aplicación del Modelo de Evaluación de Calidad	81

Tabla 55 Resultado del Análisis Comparativo	84
Tabla 56 Caso de Uso – ingresarAlSistema	90
Tabla 57 Caso de Uso – registrarUsuario.....	90
Tabla 58 Caso de Uso – buscarUsuario.....	91
Tabla 59 Caso de Uso – editarUsuario.....	92
Tabla 60 Caso de Uso – eliminarUsuario	92
Tabla 61 Caso de Uso - registrarPerfil.....	93
Tabla 62 Caso de Uso – buscarPerfil	94
Tabla 63 Caso de Uso – editarPerfil	94
Tabla 64 Caso de Uso – eliminarPerfil	95
Tabla 65 Caso de Uso – asignarPerfilAusuario	96
Tabla 66 Caso de Uso – asignarPermisoAperfil.....	97
Tabla 67 Caso de Uso – asignarMenuAperfil	97
Tabla 68 Caso de Uso – registrarCliente.....	98
Tabla 69 Caso de Uso – buscarCliente	99
Tabla 70 Caso de Uso – editarCliente	99
Tabla 71 Caso de Uso – eliminarCliente	100
Tabla 72 Caso de Uso – registrarObra	101
Tabla 73 Caso de Uso – buscarObra	101
Tabla 74 Caso de Uso – editarObra	102
Tabla 75 Caso de Uso – eliminarObra.....	103
Tabla 76 Caso de Uso – registrarHormigon	103
Tabla 77 Caso de Uso – buscarHormigon.....	104
Tabla 78 Caso de Uso – editarHormigon.....	105
Tabla 79 Caso de Uso – eliminarHormigon	106
Tabla 80 Caso de Uso – registrarElementoObra.....	107
Tabla 81 Caso de Uso – buscarElementoObra	107
Tabla 82 Caso de Uso – editarElementoObra	108
Tabla 83 Caso de Uso – eliminarElementoObra	109
Tabla 84 Caso de Uso – crearAditivo	110
Tabla 85 Caso de Uso – buscarAditivo.....	110
Tabla 86 Caso de Uso – editarAditivo	111
Tabla 87 Caso de Uso – eliminarAditivo.....	112
Tabla 88 Caso de Uso – registrarCarro	113
Tabla 89 Caso de Uso – buscarCarro	113
Tabla 90 Caso de Uso – editarCarro	114
Tabla 91 Caso de Uso – eliminarCarro.....	114
Tabla 92 Caso de Uso – registrarChofer	115
Tabla 93 Caso de Uso – buscarChofer.....	116
Tabla 94 Caso de Uso – editarChofer	116
Tabla 95 Caso de Uso – eliminarChofer.....	117
Tabla 96 Caso de Uso – registrarPedido.....	118
Tabla 97 Caso de Uso – buscarPedidos	119
Tabla 98 Caso de Uso – editarPedido	119
Tabla 99 Caso de Uso – eliminarPedido	120
Tabla 100 Caso de Uso – visualizarPedido	121
Tabla 101 Caso de Uso – crearInspeccion.....	121
Tabla 102 Caso de Uso – buscarInspeccion	122
Tabla 103 Caso de Uso – registrarDatosInspeccion	123
Tabla 104 Caso de Uso – aprobarInspeccion.....	123
Tabla 105 Caso de Uso – cambiarFechaInspeccion	124
Tabla 106 Caso de Uso – imprimirInspecciones	125
Tabla 107 Caso de Uso – crearOrdenDePedido	126
Tabla 108 Caso de Uso – imprimirOrdenDePedido	126
Tabla 109 Caso de Uso – buscarOrdenesDePedido.....	127
Tabla 110 Caso de Uso – visualizarOrdenDePedido	128
Tabla 111 Caso de Uso – editarOrdenDePedido	128
Tabla 112 Caso de Uso – eliminarOrdenDePedido.....	129

Tabla 113 Caso de Uso – crearGuiasDeRemision	130
Tabla 114 Caso de Uso – visualizarGuiaDeRemision	130
Tabla 115 Caso de Uso – editarGuiaDeRemision	131
Tabla 116 Caso de Uso – anularGuiaDeRemision	132
Tabla 117: Caso de Uso – imprimirGuiaDeRemision	133
Tabla 118 Caso de Uso – crearCilindrosLaboratorio.....	134

GLOSARIO

A

AJAX: Técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente (navegador web) mientras se mantiene la comunicación con el servidor.

API: Es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software.

ASP.net: Es un framework para aplicaciones web desarrollado y comercializado por Microsoft.

B

Back-end: Es la parte que procesa la entrada de datos desde el cliente.

C

CSS: Es un mecanismo simple para agregar estilo (por ejemplo: fuentes, colores, espaciado) a los documentos web. (W3C, 2014)

CLASS: Plantilla para la creación de objetos de datos según un modelo predefinido.

C#: Es un lenguaje de programación que se ha diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework.

Core: Es la parte fundamental del framework, su importancia es única y esencial para que los frameworks funcionen.

D

DOM: (Document Object Model): Es una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

DRY (Don't Repeat Yourself): Es una filosofía de definición de procesos que promueve la reducción de la duplicación de código.

F

FRAME: Marcos o cuadros que permiten dividir la ventana en varias más, de modo que en cada una de ellas se cargue una página HTML distinta.

Front-end: Es la parte del software que interactúa con el o los usuarios.

G:

GET: Método que se usa en el desarrollo de aplicaciones web para enviar datos desde el cliente (navegador web) hacia el servidor. Los datos se envían a través de la URL.

H

HTTP: Protocolo de transferencia de hipertexto.

HTML: Lenguaje de marcas de hipertexto para la elaboración de páginas web.

J

JSON: Notación literal de objetos de JavaScript que no requiere el uso de XML.

M

MIT: Tipo de licencia para distribuir software. No existe limitación para usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar y/o vender copias del Software. (Radcliffe, 2014)

N

Namespace: Permite agrupar constantes, clases y funciones para definir las como librerías con un nombre específico.

P

POST: Método que se usa en el desarrollo de aplicaciones web para enviar datos desde el cliente (navegador web) hacia el servidor. Los datos se incluirán en el cuerpo de la petición.

PHP: Es un lenguaje de programación de uso general de código del lado del servidor, originalmente diseñado para el desarrollo web de contenido dinámico.

POO: Programación Orientada a Objetos.

U

UML: Lenguaje Unificado de Modelado. Es un lenguaje de modelado de sistemas de software.

UWE: Metodología para el desarrollo de software orientado a la web, se enfoca en la creación de sistemas adaptativos.

URL: Localizador de Recursos Uniforme. Es una secuencia de caracteres que representa una dirección para localizar un recurso en la web.

URI: Identificador Uniforme de Recursos. Es una cadena corta que identifica un recurso, los mismos que son accesibles en una red o sistema.

utf-8: Es un formato de codificación de caracteres Unicode e ISO 10646 utilizando símbolos de longitud variable.

X

XML: Lenguaje de Marcas Extensible. Es un lenguaje que se utiliza para almacenar datos en forma legible.

XHTML: Es básicamente HTML expresado como XML válido, su objetivo es separar la información y la forma de presentarla.

XMLHttpRequest: También referida como XMLHttpRequest (Extensible Markup Language / Hypertext Transfer Protocol), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web.

RESUMEN

Este proyecto se enfoca en la elección de un framework JavaScript (jQuery o MooTools) que se utilizará como soporte a la implementación de aplicaciones Web de la empresa SOFYA SISTEMAS. Para elegir el framework que brinde las mejores características técnicas se desarrollará un modelo de evaluación compuesto por factores de calidad, basados en la norma ISO/IEC 25000 y métricas de evaluación de cumplimiento de los frameworks. En la creación del modelo, se empleará el método IQMC como guía para la identificación de características, subcaracterísticas, atributos derivados y atributos básicos apropiados para la evaluación del software. El modelo se desarrollará haciendo énfasis en la flexibilidad, escalabilidad y reutilización, con la finalidad de que este proyecto sirva para futuras evaluaciones de software dentro de la empresa. Para verificar los beneficios del framework seleccionado en base al modelo creado, se desarrollarán ciertos módulos del sistema SCPC (Sistema de Comercialización y Producción de Concreto), usando la metodología web UWE. Adicionalmente, se realizarán manuales tanto técnicos como de usuario de la herramienta seleccionada.

Palabras clave: MODELO DE EVALUACIÓN, FRAMEWORKS JAVASCRIPT, IQMC, ISO/IEC 25000, UWE.

ABSTRACT

This project focuses on choosing a JavaScript framework (jQuery or MooTools) to be used as support for the implementation of Web applications SOFYA SYSTEMS company. To choose the framework that provides the best technical features an evaluation model consisting of quality factors based on ISO / IEC 25000 and metrics for evaluating compliance with the standard frameworks will be developed. In creating the model, IQMC method shall be used as a guide for identifying features, subfeatures derivatives appropriate attributes and basic attributes for evaluating the software. The model will be developed with an emphasis on flexibility, scalability and reusability, in order that this project will serve for future evaluations of software within the company. To verify the benefits of the selected framework based on the model created, certain system modules SCPC (Sistema de Comercialización y Producción de Concreto) was developed, using the web method UWE. Additionally, technicians perform both manual and user of the selected tool .

Keywords: ASSESSMENT MODEL, JAVASCRIPT FRAMEWORKS, IQMC, ISO/IEC 25000, UWE

CAPÍTULO 1:

ANÁLISIS COMPARATIVO DE FRAMEWORKS JAVASCRIPT: JQUERY Y MOOTOOLS, PARA LA IMPLEMENTACIÓN DE APLICACIONES WEB EN LA EMPRESA SOFYA. APLICACIÓN A UN CASO DE ESTUDIO.

1.1. ANTECEDENTES

SOFYA SYSTEMS S.A. es una empresa de tecnología, dedicada al desarrollo de aplicaciones orientadas a la Web. Se estableció en el mercado el 04 de Enero del 2013, su objetivo es desarrollar software de calidad para satisfacer las necesidades de los clientes.

Actualmente la organización está implantando parámetros para la estandarización de procesos de desarrollo de software. Usa la metodología UWE para gestionar tanto el dominio como las decisiones de diseño del software a desarrollar.

A pesar de los estándares y mejoras que la organización está implantando, cabe mencionar que la empresa SOFYA no ha identificado un framework que permita desarrollar software de forma ágil.

“La elevada complejidad de muchas de las aplicaciones informáticas de hoy en día hace prácticamente inviable el desarrollo de aplicaciones sin mecanismos de reutilización, los cuales permitan a los programadores evitar partir desde cero en cada proyecto. Para ello, una solución ampliamente extendida es la utilización de Frameworks de desarrollo”. (Morcillo, 2010)

1.2. PLANTEAMIENTO DEL PROBLEMA

Actualmente la empresa auspiciante tiene retrasos en la producción, debido a:

- Trabajo desorganizado entre los desarrolladores.
- Creación de componentes que en frameworks JavaScript ya existen.

Lo mencionado anteriormente produce inconvenientes con los clientes, desventaja frente a la competencia y pérdidas económicas.

El Director Técnico de la empresa considera que los citados problemas se ocasionan debido a que no se utiliza un framework para el desarrollo de software, también ha tomado en cuenta que cualquier cambio de herramientas para el desarrollo reduce la productividad en la fase inicial; por lo tanto, tomar la decisión de la herramienta que se utilizará es de vital importancia para la organización.

Debido a lo citado anteriormente, se ha previsto en el presente trabajo realizar la evaluación y selección de un framework que brinde las mejores características para desarrollar software de calidad y de manera ágil.

1.3. JUSTIFICACIÓN

“En la actualidad existen frameworks JavaScript que proveen funcionalidades genéricas que pueden ser utilizadas para desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, optimizando tiempo y esfuerzo” (Tavárez, 2009)

Dados los problemas antes mencionados y tomando en cuenta las referencias expuestas, el presente trabajo de graduación se plantea en relación a la necesidad de establecer un framework como estándar dentro de la empresa SOFYA, con el fin de facilitar y agilizar el desarrollo de software.

Para seleccionar el framework que más se ajuste a las necesidades de la organización y brinde las mejores ventajas en el desarrollo de software, se han tomado en cuenta dos de los mejores frameworks JavaScript: jQuery y MooTools (APPERS, 2008).

1.4. OBJETIVOS

1.4.1. GENERAL

Realizar el análisis comparativo de Frameworks JavaScript: jQuery y MooTools, para la implementación de aplicaciones Web en la empresa SOFYA. Aplicación a un caso de estudio.

1.4.2. ESPECÍFICOS

- Construir un modelo de evaluación, utilizando IQMC (Método de Construcción de Calidad Individual), considerando los atributos que indica la Norma ISO 25000.
- Seleccionar el framework en base al modelo de evaluación construido.
- Aplicar el framework en el desarrollo del SCPC (Sistema de Comercialización y Producción de Concreto).
- Desarrollar manuales técnicos y de usuario del framework seleccionado.

1.5. ALCANCE

El presente trabajo de graduación, empieza con la construcción de un modelo de evaluación, el cual proporcionará una base para evaluar la calidad de los frameworks seleccionados (jQuery y MooTools), dicho modelo se construirá con IQMC (Método de Construcción de Calidad Individual), considerando los atributos de calidad que indica la Norma ISO 25000.

Para identificar el framework que brinde las mejores características de calidad (Funcionalidad, Rendimiento, Compatibilidad, Usabilidad, Fiabilidad, Seguridad, Mantenibilidad y Portabilidad) para el soporte a la implementación de aplicaciones Web, se aplicará el modelo de evaluación desarrollado.

La aplicación se implementará con soporte del framework seleccionado y ésta contará con los siguientes módulos:

- Gestión de Entidades: Crear, buscar, editar y eliminar: usuarios, perfiles, clientes, obras, hormigones, elementos de obra, aditivos, carros, choferes.
- Programación: Gestión de una agenda para la creación de pedidos de concreto.
- Inspección: Registrar datos de inspecciones realizadas en las obras donde se entrega el concreto.
- Orden de Pedido: Después que los pedidos han sido inspeccionados y aprobados se generan las órdenes de pedidos para que puedan ser despachados posteriormente.
- Despacho: En este proceso el operador divide la cantidad total de hormigón en cantidades parciales para luego ser transportadas por determinados carros a la obra especificada.

El proyecto finaliza con la creación de manuales técnicos y de usuario del framework seleccionado.

CAPÍTULO 2: MARCO TEÓRICO DE REFERENCIA

2.1. USO DE FRAMEWORKS JAVASCRIPT.

2.1.1. INTRODUCCIÓN A LOS FRAMEWORKS.

Los Frameworks son una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Son diseñados con la intención de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. (Osmani, 2012)

2.1.2. RAZONES PARA UTILIZAR UN FRAMEWORK JAVASCRIPT

La razón más importante es disminuir la inversión de recursos como el tiempo y dinero a la hora de crear funcionalidades específicas para una aplicación web. Los componentes más comunes e importantes para el desarrollo de aplicaciones web requieren escribir extensas líneas de código JavaScript pero jQuery o MooTools simplifican este trabajo debido a las funciones pre elaboradas que poseen en sus librerías.

Otra de las razones es la flexibilidad y extensibilidad que brindan estas herramientas a través de una API fácil de usar que funciona a través de una multitud de navegadores.

En la Figura 1 se muestra un ejemplo en el que se puede observar que un framework JavaScript ayuda a escribir menos líneas de código al momento de crear un determinado componente.

Ejemplo: Petición ajax - jQuery

```

1 $.ajax({
2   url: 'server.php',
3   type: "POST",
4   data: {
5     numero1: 10,
6     numero2: 20
7   },
8   success: function getResponseText(text) {
9     alert(text);
10  }
11 });

```

Resultado:

Ejemplo: Petición ajax - Javascript

```

1 cargarContenido();
2 function cargarContenido() {
3   var numero1 = 10;
4   var numero2 = 20;
5   var ajax = nuevoAjax();
6   ajax.open('POST', 'server.php', true);
7   ajax.onreadystatechange = function() {
8     if (ajax.readyState == 4) {
9       alert(ajax.responseText);
10    }
11  };
12  ajax.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
13  ajax.send('numero1=' + numero1 + '&numero2=' + numero2);
14 }
15 function nuevoAjax() {
16   var xmlhttp = false;
17   try {
18     xmlhttp = new XMLHttpRequest('Maxml2.XMLHTTP');
19   } catch (e) {
20     try {
21       xmlhttp = new XMLHttpRequest('Microsoft.XMLHTTP');
22     } catch (E) {
23       xmlhttp = false;
24     }
25   }
26   if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
27     xmlhttp = new XMLHttpRequest();
28   }
29   return xmlhttp;
30 }

```

Resultado:

Figura 1 Petición Ajax, comparación entre Framework jQuery y JavaScript.

Fuente: (jQuery, 2014)

2.2. FRAMEWORK jQuery

2.2.1. INTRODUCCIÓN

jQuery es un framework JavaScript rápido, simple y liviano que cuenta con una API fácil de usar y que funciona a través de una multitud de navegadores con una combinación de versatilidad y capacidad de ampliación.

2.2.2. CARACTERÍSTICAS

- Utiliza el paradigma de programación que usa JavaScript: “Orientado a prototipos”.
- Se enfoca en la manipulación y recorrido de elementos DOM.
- Hace posible los efectos y animaciones en una página web.
- Soporta extensiones.
- Posee utilidades para obtener información del navegador, operar con objetos y vectores, etc.
- Compatible con los navegadores Mozilla Firefox, Internet Explorer, Safari, Opera y Google Chrome.
- Su distribución se realiza bajo la licencia MIT.
- Extensa comunidad de desarrollo.
- De fácil aprendizaje.
- Interacción con el servidor a través de Ajax.
- Personalización de Efectos y Animaciones.

2.2.3. POO. BASADA EN PROTOTIPOS

jQuery no hace énfasis en la programación modular, flexible y reutilizable. Este framework se enfoca en simplificar el recorrido del DOM, manejo de eventos, animación, etc. Es por esto que si se requiere de las bondades de la programación orientada a objetos se utiliza JavaScript (Lenguaje de Programación Orientado a Objetos basado en Prototipos). Según los autores del sitio web de Mozilla, la utilización de prototipos en vez de clases no limita al lenguaje, todas las características como herencia, encapsulamiento, abstracción y polimorfismo se pueden obtener sin problemas con JavaScript.

En la Figura 2 y 3 se indica la diferencia que existe entre la POO. basada en prototipos y la POO. basada en clases.

Programación Orientada a Objetos basada en Prototipos

```
var Persona = function(nombre, apellido, edad) {
  //Atributos
  this.nombre = new String(nombre);
  this.apellido = new String(apellido);
  this.edad = new Number(edad);
  //Método
  this.mostrarDatos = function() {
    return "Nombre: " + this.nombre + ", Apellido: " + this.apellido + ", Edad: " + this.edad;
  };
};
```

Clonación

```
$(function() {
  var personal = new Persona("Juan", "Basantes", 34);
  $('#datosPersona').text(personal.mostrarDatos());
});
```

Resultado

Nombre: Juan, Apellido: Basantes, Edad: 34

Figura 2 Programación Orientada a Objetos basada en Prototipos

Fuente: (jQuery, 2014)

Programación Orientada a Objetos basada en Clases

```
public class Persona {
  //Atributos
  private String nombre;
  private String apellido;
  private Integer edad;
  //Método
  public String mostrarDatos(){
    return "Nombre: " + this.nombre + ", Apellido: " + this.apellido + ", Edad: " + this.edad;
  }
}
```

Instanciación

```
public static void main(String[] args) {
  Persona personal = new Persona("Juan", "Basantes", 34);
  System.out.println(personal.mostrarDatos());
}
```

Resultado

Nombre: Juan, Apellido: Basantes, Edad: 34

Figura 3 Programación Orientada a Objetos basada en clases.

Fuente: (jQuery, 2014)

2.2.4. AJAX

jQuery posee un conjunto completo de funcionalidades para realizar peticiones Ajax. Estas funcionalidades se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible cargar datos en las páginas sin la necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad.

Funcionalidades:

- La técnica Ajax ejecuta una función predeterminada:
 - Cuando una petición al servidor se completa.
 - Cuando una petición al servidor se completa con errores.
 - Antes que una petición se envíe al servidor.
 - Cuando se inicia la primera petición.
 - Cuando todas las peticiones se completan.
 - Cuando una petición se ha completado con éxito.
- Gestiona parámetros antes de que estos sean procesados y enviados al servidor.
- Establece valores predeterminados para las futuras solicitudes.
- Crea un objeto que se encarga de la transmisión de datos (Si la respuesta del servidor no es la esperada, el objeto envía otros valores o aborta la comunicación).
- Envía datos al servidor mediante una petición HTTP GET.
- Recibe datos del servidor codificados en formato JSON mediante una petición HTTP GET.
- Carga un archivo JavaScript desde el servidor mediante una petición HTTP GET.
- Crea una representación serializada de una matriz u objeto, adecuada para su uso en una cadena de consulta URL o solicitud Ajax.
- Carga datos del servidor mediante una petición HTTP POST.
- Carga datos del servidor con formato HTML.

- Codifica un conjunto de elementos de formularios como una cadena de texto para su presentación.
- Codifica un conjunto de elementos de formulario como una matriz de nombres y valores.

2.2.5. MANIPULACIÓN DEL DOM

jQuery implementa una serie de métodos que manipulan el DOM HTML y XHTML, se categorizan por la manipulación de:

- Elementos HTML: Obtiene y Establece elementos en una página web.
- Atributos de un elemento HTML: tipo, nombre, id, class.
- Aspecto y Formato de un elemento HTML: Propiedades de fuente, color, dimensiones, texturas, etc.
- Datos: Algunos de estos métodos también actúan como captadores (recuperan información de elementos DOM para su uso posterior).

2.2.6. VALIDACIÓN DE FORMULARIOS

jQuery cuenta con una serie de librerías orientadas a la validación de los campos de un formulario, estas funcionan en los siguientes navegadores: Internet Explorer, Chrome, Firefox, Safari y Opera, cada librería posee una amplia gama de funciones que brindan al desarrollador la posibilidad de validar texto y acciones que realizan los usuarios, implementando líneas de código simplificadas.

Los mensajes que se muestran cuando se manipula un formulario son visualmente atractivos y se los puede personalizar a gusto del desarrollador.

Las validaciones que se realizan en los formularios van desde correo electrónico, teléfono, dirección URL hasta solicitudes complejas tales como el procesamiento de Ajax o funciones JavaScript.

Los mensajes de error se pueden personalizar y configurar para varios idiomas.

2.2.7. EFECTOS VISUALES

jQuery posee una biblioteca (jQuery UI) dedicada especialmente al desarrollo de aplicaciones amigables para el usuario, la librería posee un conjunto de funciones que añaden los siguientes comportamientos complejos a los elementos HTML:

- Arrastrar.
- Arrastrar y Soltar.
- Redimensionar.
- Ordenar.
- Responder a elementos que se pueden arrastrar.

jQuery UI incorpora un conjunto completo de elementos dinámicos, cada elemento posee opciones configurables como la posibilidad de aplicar estilos CSS. Los elementos son:

- Accordion: Menú con efecto acordeón.
- Autocomplete: Caja de texto con autocompletado.
- Button: Botón.
- Dialog: Ventanas con contenido.
- Slider: Elemento para elegir en un rango de valores.
- Tabs: Pestañas.
- Datepicker: Calendario gráfico.
- Progressbar: Barra de progreso.

La biblioteca también incorpora efectos visuales que permiten construir aplicaciones web altamente interactivas, entre éstos se tiene:

- Blind.
- Bounce.
- Clip.
- Drop, entre otros 10 efectos.

2.2.8. MANEJO DE EVENTOS

jQuery también incorpora una serie de eventos, los cuales son representados como métodos que se utilizan para registrar las conductas que se aplican cuando el usuario interactúa con el navegador y para manipular esos comportamientos registrados.

En la actualidad hay 58 eventos a disposición de los usuarios, éstos aparte de responder a una determinada acción, gestionan o controlan las conductas de ciertas funcionalidades como por ejemplo el manejo de errores JavaScript.

2.2.9. jQuery PARA MÓVILES

jQuery posee una librería especializada para la creación de interfaces de usuario basadas en HTML5, estas son diseñadas para que los sitios y aplicaciones web puedan ser visitadas por cualquier dispositivo móvil (Smartphone, tablet, laptop).

Con jQuery Mobile, en lugar de escribir aplicaciones únicas para cada dispositivo móvil o sistema operativo, permite crear y diseñar sitios web independientes de marcas, esto quiere decir que el software desarrollado con jQuery Mobile puede trabajar en todas las plataformas de teléfonos inteligentes, tabletas o laptops más populares.

Dentro de las funcionalidades que posee la librería se puede encontrar "ThemeRoller", función que permite construir temas personalizados de una manera fácil e intuitiva.

La librería tiene compatibilidad con los siguientes dispositivos:

- Apple iOS 4-7.0: Probado en iPad (4.3 / 5.0), iPad 2 (4.3 / 5.1 / 6.1), iPad 3 (5.1 / 6.0), iPad Mini (6.1), iPad Retina (7.0), iPhone 3GS (4.3), iPhone 4 (4.3 / 5.1), iPhone 4S (5.1 / 6.0), iPhone 5 (6.0), y iPhone 5S (7.0)
- Android 4.4 (KitKat): Probado en Nexus 5.

- Android 4.1-4.3 (Jelly Bean): Probado en Galaxy Nexus y Galaxy 7.
- Android 4.0 (ICS): Probado en Galaxy Nexus.
- Android 3.2 (Honeycomb): Probado en Samsung Galaxy Tab 10.1 y Motorola XOOM.
- Android 2.1-2.3: Probado en HTC Incredible (2.2), original Droid (2.2), HTC Aria (2.1), Google Nexus S (2.3).
- Windows Phone 7.5-8: Probado en HTC Surround (7.5), HTC Trophy (7.5), LG-E900 (7.5), Nokia 800 (7.8), HTC Maza (7.8), Nokia Lumia 520 (8), Nokia Lumia 920 (8), HTC 8x (8).
- Blackberry 6-10: Probado en Torch 9800 (6), Style 9670 (6), BlackBerry® Torch 9810 (7), BlackBerry Z10 (10).
- BlackberryPlaybook (1.0-2.0): Probado en PlayBook.
- Palm WebOS (1.4-3.0): Probado en Palm Pixi (1.4), Pre (1.4), Pre 2 (2.0), HP TouchPad (3.0).
- Firefox Mobile 18: Probado en dispositivos Android 2.3 y 4.1.
- Chrome para Android 18: Probado en dispositivos Android 4.0 y 4.1.
- Skyfire 4.1: Probado en dispositivos con Android 2.3.
- Opera Mobile 11.5-12: Probado en Android 2.3.
- MeeGo 1.2: Probado en Nokia 950 y N9.
- Samsung Bada 2.0: Probado en Samsung Wave 3, Dolphin browser
- UC Browser: Probado en dispositivos con Android 2.3.
- Kindle 3, Fire, and Fire HD: Probado en navegadores con WebKit.
- Nook Color 1.4.1: Probado en Nook Color, not Nook Tablet.
- Chrome Desktop 16-24: Probado en OS X 10.7 y Windows 7.

- Safari Desktop 5-6: Probado en OS X 10.8.
- Firefox Desktop 10-18: Probado en OS X 10.7 and Windows 7.
- Internet Explorer 8-10: Probado en Windows XP, Vista y 7, Windows Surface RT.
- Opera Desktop 10-12: Probado en OS X 10.7 y Windows 7.
- Opera Mini 7: Probado en iOS 6.1 y Android 4.1.
- Nokia Symbian^3: Probado en Nokia N8 (Symbian^3), C7 (Symbian^3), también trabaja en N97 (Symbian^1).
- Internet Explorer 7 y versiones anteriores: Probado en Windows XP.
- Apple iOS 3.x y versiones anteriores: Probado en iPhone (3.1), iPhone 3 (3.2).
- Blackberry 4-5: Probado en Curve 8330 (4), Storm 2 9550 (5), y Bold 9770 (5)
- Windows Mobile: Probado en HTC Leo (WinMo 5.2).

2.2.10. PRUEBAS UNITARIAS

QUnit es un framework potente y fácil de usar que sirve para realizar pruebas unitarias de código JavaScript. Es utilizado por los proyectos de jQuery, jQuery UI y jQuery Mobile, también es capaz de probar cualquier código genérico JavaScript, incluyendo funcionalidades del mismo framework.

QUnit fue desarrollado en el 2008 por John Resig (Creador de jQuery). Originalmente fue escrito como una librería perteneciente al framework jQuery, pero desde el 2009 QUnit funciona de forma independiente.

2.2.11. INSTALACIÓN DE jQuery

1. Acceder a <http://jquery.com/download/> para descargar la última versión del framework. Existen dos tipos de librerías para descargar:

- a. PRODUCTION: adecuada para páginas web en producción, puesto que está minimizada y ocupa menos espacio, con lo que la ejecución de un sitio web será más rápida.
 - b. DEVELPOMENT: el código está sin compresión, ocupa más espacio, pero se podrá leer la implementación de las funciones del framework.
2. Copiar el archivo descargado ".js" en un directorio para luego referenciarlo.
 3. En un archivo ".html" entre las etiquetas "<head> y </head>" copiar la siguiente referencia: "<script src='directorioX/query-1.3.2.min.js' type='text/javascript'></script>", donde:
 - a. directorioX: Carpeta donde se guarda la librería.
 - b. query-1.3.2.min.js: Nombre del framework.

2.2.12. INFORMACIÓN TÉCNICA

Tabla 1

Información técnica del Framework jQuery

No.	Framework	jQuery
1	Última versión	2.1.0
	Tamaño de la Versión	81,6 KB
2	"PRODUCTION"	
	Tamaño de la Versión	239 KB
3	"DEVELOPMENT"	
4	Licencia	MIT
5	Año Inicio	Enero 2006
	Cross-Browser	Safari 6.0+, Internet Explorer 9.0+, Firefox
6	(Compatibilidad Navegadores)	27+, Opera 12.1x, and Chrome 33+.

Fuente: (jQuery, 2014)

2.2.13. VERSIONES DE jQuery

Los desarrolladores de jQuery ponen a disposición de los usuarios versiones de las librerías organizadas en las siguientes categorías:

Tabla 2
Versiones del Framework jQuery

No.	Categoría	Descripción	Versiones disponibles
1	jQuery	Librería para la manipulación del DOM, uso de Ajax, eventos, etc.	111
2	jQuery UI	Librería con componentes prefabricados para el desarrollo de aplicaciones web.	43
3	jQuery Mobile	Librería para el desarrollo de componentes adaptables a dispositivos móviles.	38
4	jQuery Color	Librería para el desarrollo de temas para aplicaciones web.	8
5	QUnit	Librería para realizar pruebas unitarias.	11

Fuente: (jQuery, 2014)

2.2.14. COMUNIDAD

El framework está dirigido oficialmente por un grupo de 45 personas que hacen posibles mejoras y nuevas actualizaciones sobre la herramienta. jQuery también cuenta con el apoyo de miembros corporativos como son: WordPress, MaxCDN, mt (midiatemple), Intel, 10up, entre otros veinte y dos integrantes.

Una ventaja de esta herramienta es que existe una gran cantidad de colaboradores que han desarrollado componentes disponibles en la web con licencias gratuitas y algunas privativas.

2.3. FRAMEWORK MooTools

2.3.1. INTRODUCCIÓN

MooTools es un framework JavaScript compacto, modular y orientado a objetos, fue diseñado para desarrolladores de software de nivel intermedio y avanzado. La herramienta permite escribir código flexible, potente y portable gracias a una API coherente, elegante y bien documentada.

2.3.2. CARACTERÍSTICAS

- Se compone de diversos módulos, se puede seleccionar sólo los que se ocuparán en las páginas web, dejando los otros para que no ocupen tiempo de descarga ni procesamiento.
- Es orientado a Objetos basado en Clases.
- Enfatiza la extensión, herencia, legibilidad, reusabilidad y mantenibilidad de código.
- Se enfoca en hacer la API de JavaScript más estable y coherente.
- Sigue los principios DRY, que hacen de él un framework potente y eficiente.
- Implementa la función "Class" que hace que el modelo de herencia prototipada de JavaScript sea más sencillo.
- Posee un conjunto de componentes, animaciones y efectos que hacen dinámica a una página web.
- Interacción con el servidor a través de Ajax.
- Su distribución se realiza bajo la licencia MIT.
- Manipula y recorre elementos DOM de un página HTML.

- Emula código jQuery.
- Compatible con los navegadores Safari, Internet Explorer, Mozilla Firefox, Opera y Google Chrome.

2.3.3. POO. BASADA EN CLASES

MooTools permite realizar programas o scripts con programación orientada a objetos basada en clases. Provee una serie de utilidades para crear clases de una manera más simplificada y elegante. En la Figura 4 se muestra la Sintaxis utilizada para la creación de una clase.

```
var MiClase = new Class(propiedades);
```

Figura 4 Sintaxis MooTools para instanciar una clase

Fuente: (MooTools, 2014)

Argumentos: Son propiedades que pueden ser de dos tipos:

- Objeto: Colección de propiedades que se aplican a la clase, también acepta propiedades especiales como Extends (En el caso de herencia) e Implements (Simulación de herencia múltiple).
- Función: Initialize (Constructor de la Clase), toElement (Un método que devuelve un elemento. Este método será llamado automáticamente cuando pasa una instancia de una clase).

Datos de Retorno: La Clase creada.

La Figura 5 , indica un ejemplo de la función Class que usa el Framework JavaScript.

```
var Animal = new Class({
  initialize: function(edad){
    this.age = edad;
  }
});

var Gato = new Class({
  Extends: Animal,
  initialize: function(nombre, edad){
    this.parent(edad);
    this.name = nombre;
  }
});

var objGato = new Gato('Micia', 3);
```

Figura 5 Ejemplo de uso de Clases en MooTools

Fuente: (MooTools, 2014)

2.3.4. AJAX

Una de las mejores utilidades de cualquier framework JavaScript es la posibilidad de crear aplicaciones web utilizando el mecanismo Ajax para establecer una comunicación entre el cliente y el servidor, mostrando datos en la aplicación sin tener que recargar todo el contenido de la páginas.

Programar scripts que utilizan Ajax es un poco complicado porque cada navegador tiene sus particularidades para producir las llamadas al servidor y recibir una respuesta, lo que obliga en un principio a realizar scripts con distinto código a ejecutar dependiendo del cliente que visita nuestro sitio web, pero con MooTools no hay problema de que navegador está utilizando el visitante porque el framework dispone de clases con métodos que funcionan de la misma manera en todos los navegadores Web. (Álvarez, 2010).

MooTools tiene una clase "Request" que posee opciones de configuración, eventos y métodos necesarios para realizar una conexión Ajax desde el cliente.

Opciones de configuración:

- url (Tipo de dato: String): Dirección de la petición, también puede ser una instancia de URI.

- data (Tipo de dato: mixed): Datos que se envía desde el cliente hacia el servidor, cuando no se especifica el o los datos, el valor por defecto es "send".
- format (Tipo de dato: String): Si se envía un identificador de formato adicional, este puede ser agregado a la opción "data", ejemplo: '&format=json'.
- link (Tipo de dato: String): Pueden ser los siguientes valores:
 - 'ignore': Cualquier petición realizada es ignorada.
 - 'cancel': Cualquier petición realizada que no sea la actual, tiene prioridad mientras esta tenga el link en 'cancel'.
 - 'chain': Si existen varias peticiones por el cliente, estas se van ejecutando una tras otra, dependiendo el orden en el que fueron ejecutadas.
- method (Tipo de dato: String): Tipo de petición, puede ser 'POST' o 'GET'.
- emulation (Tipo de dato: Booleano): Si se establece en true, otros métodos 'POST' o 'GET'.
- se anexan con nombre '_method'.
- async (Tipo de dato: Booleano): Si el valor es true, la petición se realiza de forma asíncrona, caso contrario de forma síncrona.
- timeout (Tipo de dato: Integer): determina la cantidad de milisegundos antes de considerar una conexión de tiempo de espera.
- headers (Tipo de dato: Object): Objeto creado con el fin de establecer los encabezados de solicitud.
- urlEncoded (Tipo de dato: Booleano): Si se establece en true, el encabezado de tipo de contenido se establece en "urlencoded-www-form + codificación".
- encoding (Tipo de dato: Booleano): Por defecto 'utf-8', .
- noCacheb (Tipo de dato: Booleano): Parámetro para evitar el almacenamiento en caché.
- isSuccess (Función): Define una función que es usada cuando una petición es satisfactoria.

- evalScripts (Tipo de dato: Booleano): Si se establece en true, se evaluarán etiquetas <script> dentro de la respuesta del servidor.
- evalResponse (Tipo de dato: Booleano): Si se establece en true, se evaluará toda la respuesta del servidor.
- user (Tipo de dato: String): Cuando se establece el nombre de usuario, la solicitud se completará si la autenticación fue satisfactoria.
- password (Tipo de dato: String): Esta opción se usa junto con user.

Eventos:

- request: Se ejecuta cuando la petición ha sido enviada.
- loadstart: Se ejecuta cuando una solicitud se ha ejecutado.
- progress: Se ejecuta cuando la solicitud está haciendo procesos de descarga.

Funciones:

- complete: Se ejecuta cuando la petición ha sido completada.
- cancel: Se ejecuta cuando una petición ha sido cancelada.
- success: Se ejecuta cuando una petición ha sido completada satisfactoriamente.
- failure: Se ejecuta cuando una petición falla.
- exception: Se ejecuta cuando la configuración de la cabecera de una petición tiene errores en la parametrización.

2.3.5. MANIPULACIÓN DEL DOM

Una de las clases más importantes que tiene Mootools es "Element" la cual brinda métodos para manipular el DOM de páginas web de manera muy sencilla.

Mootools hace uso de dos operadores para seleccionar elementos, el signo de moneda “\$” junto a una cadena que ayuda a indicar el “id” de un elemento; mientras que para lograr la selección de una serie de elementos utiliza dos signos de moneda “\$\$” y algún selector que englobe todos los elementos que se desee. Una vez seleccionados los elementos, el framework permite realizar varias operaciones con ellos, por ejemplo: cambiar el color, modificar el tamaño, agregar valores a las propiedades de los elementos HTML, etc.

El framework permite mejorar el rendimiento de una aplicación usando concatenación de métodos y guardado en variables, de esa manera los elementos obtenidos quedan almacenados en cache y permite hacer un uso rápido de ellos.

2.3.6. VALIDACIÓN DE FORMULARIOS

MooTools tiene registrado en su sitio web 3 librerías para validar formularios, la más descargada muestra la siguiente descripción: ValidateSimple es una clase MooTools que funciona en tiempo real, valida ingresos de texto e interacciones en formularios, ejecuta eventos validadores mientras el usuario está escribiendo, también muestra mensajes que alertan al usuario de errores y aciertos que ocurren en un determinado formulario. (Valerio Proietti, 2006)

La librería permite especificar qué entradas se validan y mediante qué criterios se hacen, estos criterios son configurables en la clase (se pueden agregar o suprimir), por defecto se establecen ciertos criterios para empezar una validación.

Esta librería no es un componente con todas las funciones necesarias. Este código no aplica para configurar un tipo de interfaz diferente al proporcionado, sólo desencadena eventos y permite al usuario modificar la semántica de los mensajes que se muestran. Si se desea un validador más potente, también hay que usar MooTools Moreone.

2.3.7. EFECTOS VISUALES

Para el manejo de efectos visuales, MooTools incorpora una clase denominada “Fx” que fue especialmente diseñada para hacer efectos especiales para enriquecer la experiencia del usuario.

Existen clases que heredan de “Fx”, que sirven para hacer efectos de diversos tipos. Estas clases son:

- Fx.CSS: Esta clase tiene funcionalidades para interpretar estilos y clases CSS.
- Fx.Tween: Una clase que sirve para hacer una transición de cualquier propiedad CSS de un valor a otro. Con ella se puede hacer efectos con cualquier duración, de modo que su presentación sea lenta o rápida.
- Fx.Morph: Esta clase es parecida a Fx.Tween, con la diferencia que permite alterar varias propiedades CSS al mismo tiempo.
- Fx.Transitions: Creada para hacer efectos más detallistas que ajustan el proceso de transición con una función matemática. Por ejemplo, esto sirve para hacer un desplazamiento de un elemento, donde la velocidad del movimiento no es lineal sino exponencial siendo más rápido al inicio o al final del movimiento. Existen diferentes funciones matemáticas preestablecidas para generar varios efectos sobre cualquier elemento HTML.

MooTools también posee elementos básicos para la creación rápida de un sitio web dinámico, por ejemplo:

- Fx.Elements: Permite aplicar cualquier número de estilos y transiciones a una colección de elementos.
- Fx.Accordion: Esta clase crea un grupo de elementos HTML que visualmente tienen un efecto de acordeón.
- Fx.Move: Mueve un elemento a otra ubicación (relativa a otro elemento) con una transición.

- Fx.Reveal: Gestiona transiciones de la altura, opacidad, relleno y margen.
- Fx.Scroll: Agrega una barra de desplazamiento: vertical u horizontal a cualquier elemento que tenga desbordamiento, incluyendo la ventana del navegador.
- Fx.Slide: Añade un efecto deslizante a elementos HTML, haciendo que sus contenidos se contraigan, oculten o sobrepongan.
- Fx.Sort: Reordena un grupo de artículos con una transición predeterminada por el desarrollador.
- Drag, Drag.Move, Slider y Sortables: Estas clases proporcionan funcionalidades para arrastrar, arrastrar y soltar cualquier tipo de elementos HTML, adicionalmente reaccionan a eventos del teclado y mouse.
- HtmlTable, HtmlTable.Zebra, HtmlTable.Sort, HtmlTable.Select, Tips, Spinner: Estas clases se encargan de manipular una tabla y sus elementos, así como de mostrar mensajes personalizados cuando pasamos el cursor encima de una celda.

2.3.8. MANEJO DE EVENTOS

MooTools no sólo hace uso de los eventos que vienen por defecto en JavaScript (onclick, onload, onmouseover, etc.) para agregar dinamismo a las páginas web, sino que también agrega métodos para manipular estos eventos. Los métodos son los siguientes:

- stop: Detiene un determinado evento y también ejecuta "preventDefault".
- stopPropagation: Detiene un evento que se propaga por la activación de otro.
- preventDefault: Evita la acción predeterminada de un evento.
- DOMEvent.defineKey: Este evento permite añadir un código clave a un evento.

- `DOMEvent.defineKeys`: Esta función permite añadir teclas o combinación de teclas a un determinado evento.

Las Clases que manipulan los eventos también proporcionan atributos configurables como coordenadas, destino, encadenamiento de eventos, etc.

2.3.9. MooTools PARA MÓVILES

MooTools provee un componente denominado “Mobile” que sirve para utilizarlo en el desarrollo de aplicaciones web, con interfaces adaptables a dispositivos móviles. Proporciona eventos personalizados e información útil para el navegador web que usan los Smartphone. En iOS proporciona un controlador de eventos táctil que reemplaza automáticamente todos los controladores de clic.

El componente creado por los desarrolladores de MooTools es compatible con los siguientes dispositivos:

- iOS
 - iPhone 4 with iOS 4.0.2
 - iPad with iOS 3.2.2
 - iPod Touch 2g with iOS 4.0.1
- Android
 - HTC Magic with Android 2.2 (Cyanogenmod6): Android Browser, Dolfín HD.
 - Motorola Droid / Milestone with Android 2.1: Android Browser.
 - Nexus One with Android 2.2 (Cyanogenmod6): Android Browser.
 - HTC Desire with Android 2.2: Android Browser.
 - Samsung Galaxy S with Android 2.1: Android Browser

2.3.10. INSTALACIÓN DE MooTools

1. Acceder a <http://mootools.net/download> para descargar la última versión del framework.

Existen tres tipos de librerías para descargar:

- a. YUI Compressor: Adecuada para páginas web en producción, puesto que está comprimida y ocupa menos espacio, con lo que la carga de nuestro sitio será más rápida.
 - b. Uncompressed: El código está sin compresión, ocupa más espacio, pero se podrá leer la implementación de las funciones del framework.
 - c. Server-Side: Se trata del framework MooTools sin ningún componente relativo con la compatibilidad de los navegadores. Incluye Clase, Core y extensiones nativas. Está hecho específicamente para entornos de servidor como Node.js, v8cgi, Rhino o SpiderMonkey.
2. Copiar el archivo descargado ".js" en un directorio para luego referenciarlo.
 3. En un archivo ".html" entre las etiquetas "<head> y </head>" copiar la siguiente referencia: "<script src="directorioX/mootools-core-1.4.5-full-compat.js" type="text/javascript"></script>".donde:
 - a. directorioX: Es la carpeta donde de guarda la librería.
 - b. mootools-core-1.4.5-full-compat.js: Nombre del framework.

2.3.11. INFORMACIÓN TÉCNICA

Tabla 3

Información técnica del Framework MooTools

No.	Frameworks	Mootools
1	Última versión	1.4.5
2	Tamaño de la Versión	88 KB
	"PRODUCTION"	Continúa →

3	Tamaño de la Versión "DEVELOPMENT"	147 KB
4	Licencia	MIT
5	Año Inicio	Septiembre 2006
6	Cross-Browser (Compatibilidad de Navegadores)	Safari, Internet Explorer 6.0+, Firefox, Opera, and Chrome.

Fuente: (MooTools, 2014)

2.3.12. VERSIONES DE MOOTOOLS

Los desarrolladores de MooTools tienen disponible en su sitio web las siguientes versiones del framework:

- Librería para el desarrollo de aplicaciones web: 1.1, 1.2, 1.2.6 y 1.4.5
- Componente para móviles: 1.0.5

MooTools dispone de 2 tipos de framework para descargar: cliente y servidor. Todas las versiones están disponibles con compresión para la producción de un aplicativo y sin compresión para facilitar el desarrollo del usuario.

2.3.13. COMUNIDAD

MooTools está dirigido oficialmente por un grupo de 16 personas que hacen posibles mejoras y nuevas actualizaciones sobre la herramienta.

2.4. NORMA ISO/IEC 25000

Fuente: (ISO, 2005)

2.4.1. INTRODUCCIÓN

La norma ISO/IEC 25000 fue desarrollada por el subcomité SC7 (Ingeniería de software y sistemas), se creó con el objetivo de cubrir los

procesos de especificación de requerimientos de calidad y evaluación de calidad de software con el apoyo de procesos de medición de calidad.

La norma está constituida por una serie de normas basadas en la ISO 9126 y en la ISO 14598 (Evaluación de Software), con el objetivo principal de guiar el desarrollo de los productos software con la especificación y evaluación de requisitos de calidad; cuenta con sus propias características y subcaracterísticas de calidad.

La serie ISO 25000 no establece los niveles de calidad deseables para cada proyecto, al contrario recomienda que los requisitos de calidad deban ser proporcionales a las necesidades de la aplicación.

2.4.2. DIVISIONES DE LA NORMA ISO-25000

En la Figura 6 se muestra la división de la norma ISO/IEC 25000 y las relaciones que ésta posee.

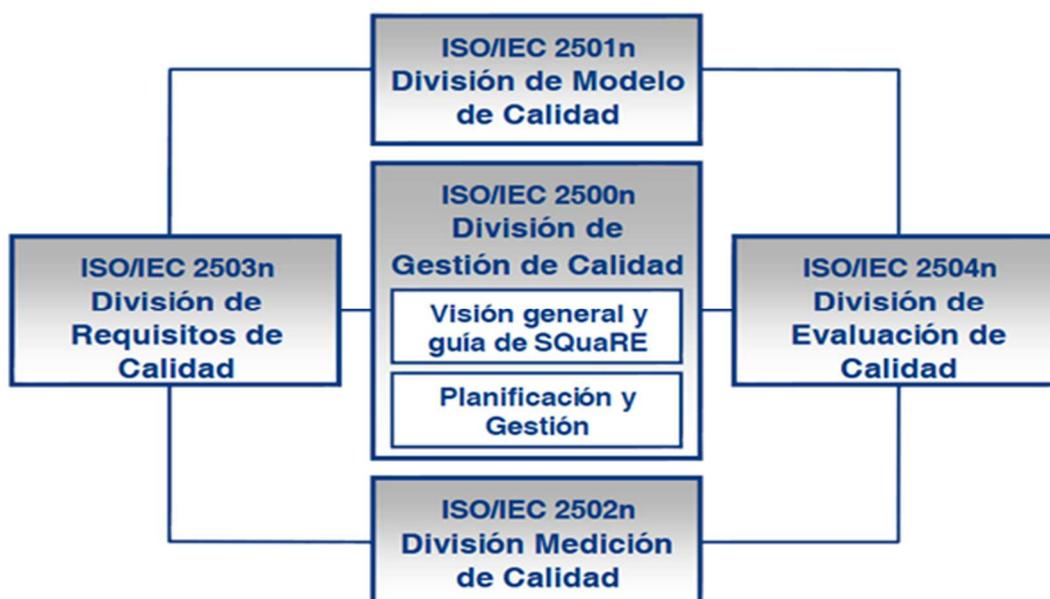


Figura 6 Divisiones de la Norma ISO/IEC 25000

Fuente: División de la Norma (ISO, 2005)

ISO/IEC 2500n. Los estándares que forman esta división definen todos los modelos comunes, términos y referencias a los que se alude en las demás divisiones de SQuaRE.

ISO/IEC 2501n. El estándar que conforma esta división presenta un modelo de calidad detallado, incluyendo características para la calidad interna, externa y en uso.

ISO/IEC 2502n. Los estándares pertenecientes a esta división incluyen un modelo de referencia de calidad del producto software, definiciones matemáticas de las métricas de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.

ISO/IEC 2503n. Los estándares que forman parte de esta división ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto software que va a ser desarrollado o como entrada para un proceso de evaluación.

ISO/IEC 2504n. Estos estándares proporcionan requisitos, recomendaciones y guías para la evaluación de un producto software, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores.

2.4.3. FAMILIA ISO/IEC 25010

Según la norma ISO 25000, el modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. El modelo de calidad está determinado por características de calidad que se tomarán en cuenta al momento de evaluar las propiedades de un producto software determinado.

En un producto software la calidad se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios. La funcionalidad, rendimiento, seguridad, mantenibilidad, etc. Son requisitos que se encuentran

representados en el modelo de calidad, el cual categoriza estos elementos en características y subcaracterísticas.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por 8 características y 31 subcaracterísticas de calidad; la Figura 7 indica la jerarquía y sus relaciones:

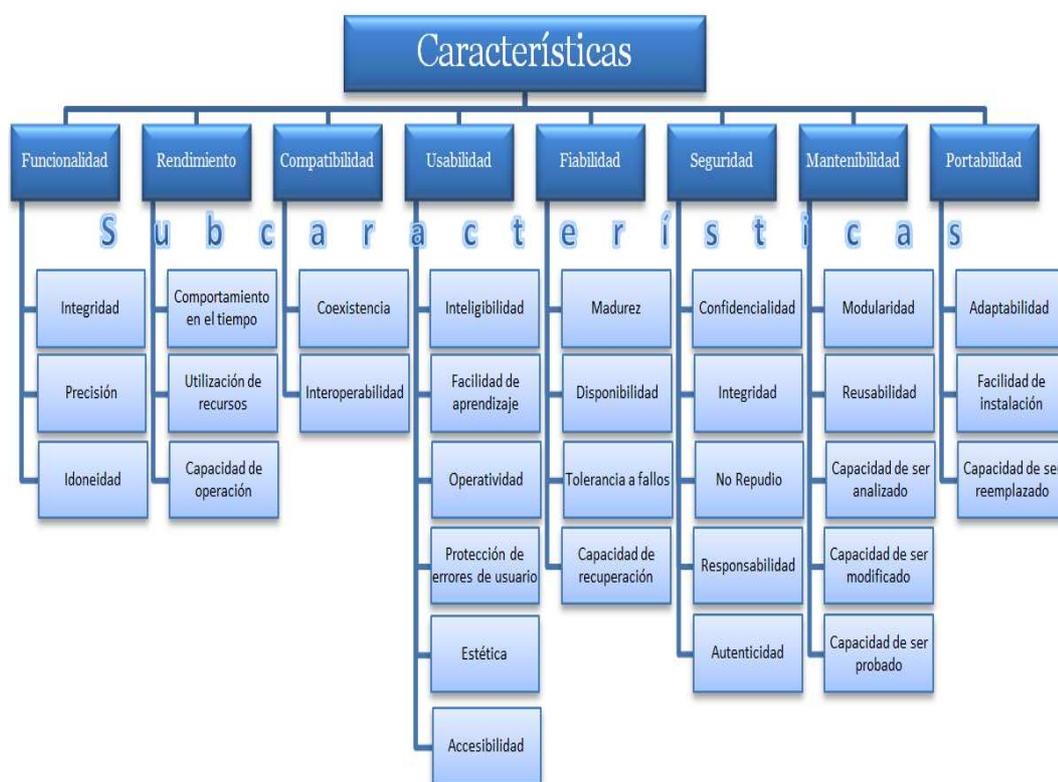


Figura 7 Características (Internas y Externas) de Calidad del Producto Software

Fuente: ISO/IEC 25010

2.4.4. FUNCIONALIDAD

Es la capacidad del producto software para proporcionar métodos que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.

División de subcaracterísticas:

- **Integridad:** Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
- **Precisión:** Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
- **Idoneidad:** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

2.4.5. RENDIMIENTO

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.

División de subcaracterísticas:

- **Comportamiento en el tiempo:** Son los tiempos de respuesta, procesamiento y ratios de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas.
- **Utilización de recursos.** Son las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
- **Capacidad de Operación.** Capacidad del producto software para adherirse a normas o convenciones relacionada con la eficiencia.

2.4.6. COMPATIBILIDAD

Es la capacidad que tienen dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software.

División de subcaracterísticas:

- **Coexistencia:** Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
- **Interoperabilidad:** Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

2.4.7. USABILIDAD

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

División de subcaracterísticas:

- **Inteligibilidad:** Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Facilidad de aprendizaje:** Capacidad del producto que permite al usuario aprender su aplicación.
- **Operatividad:** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección de errores de usuario:** Capacidad del sistema para proteger a los usuarios de hacer errores.
- **Estética:** Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- **Accesibilidad:** Capacidad del producto que permite que sea utilizado por usuarios con determinadas discapacidades.

2.4.8. FIABILIDAD

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados.

División de subcaracterísticas:

- **Madurez:** Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- **Disponibilidad:** Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- **Tolerancia a fallos.** Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- **Capacidad de recuperación:** Capacidad del producto software para recuperar los datos directamente afectados y restablecer el estado deseado del sistema en caso de interrupción o fallo.

2.4.9. SEGURIDAD

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.

División de subcaracterísticas:

- **Confidencialidad:** Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- **Integridad:** Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
- **No repudio:** Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- **Responsabilidad:** Capacidad de rastrear de forma inequívoca las acciones de una entidad.

- Autenticidad: Capacidad de demostrar la identidad de un sujeto o un recurso.

2.4.10. MANTENIBILIDAD

Representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

División de subcaracterísticas:

- Modularidad: Capacidad de un sistema (compuesto de componentes) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- Reusabilidad: Capacidad de un elemento software que permite que sea utilizado en más de un sistema o en la construcción de otros elementos.
- Capacidad de ser analizado: Facilidad con la que en un producto se puede evaluar el impacto de un determinado cambio sobre el resto del software.
- Capacidad de ser modificado: Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- Capacidad de ser probado: Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

2.4.11. PORTABILIDAD

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno software a otro.

División de subcaracterísticas:

- Adaptabilidad: Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de software, operacionales o de uso.
- Facilidad de instalación: Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- Capacidad de ser reemplazado: Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

2.5. MODELOS DE CALIDAD

Fuente: (Coral Calero, 2010)

Un modelo de calidad es el conjunto de factores y relaciones entre ellos que proporciona una base para la especificación de requisitos de calidad y para la evaluación de la calidad de los componentes software.

Pueden ser usados como base para definir un framework de calidad común y pueden ser adaptados a contextos específicos.

Proporcionan una base mensurable para la evaluación de la calidad del software.

2.5.1. TIPOS DE MODELOS DE CALIDAD

Los modelos de calidad se clasifican según su enfoque:

- Modelos de calidad fijos.
- Modelos de calidad a medida.
- Modelos de calidad mixtos.

2.5.1.1. MODELOS DE CALIDAD FIJOS

Características:

- Para realizar la evaluación de la calidad, debe existir un catálogo de factores de calidad.

- No se identifican más factores de calidad.
- No se descartan factores de calidad.
- Estructuración de factores en una jerarquía multinivel.
- Proporciona una estructura de factores común para todos los proyectos a ser evaluados.
- Poco flexibles.

Ejemplos de modelos que siguen este enfoque son los modelos de McCall (1997), Boehm (1978), Keller (1990) y el modelo con un enfoque más industrial FURPS (1987).

2.5.1.2. MODELOS DE CALIDAD A MEDIDA

Características:

- No existe ningún catálogo de factores de calidad para la evaluación.
- Los factores deben ser identificados para cada proyecto (Se debe partir de la identificación de los objetivos a alcanzar).
- Los modelos son creados desde cero para todo nuevo proyecto.
- Total adaptabilidad de los factores al modelo de evaluación.
- El coste de construcción es muy alto comparado con el de los modelos fijos.
- No es aplicable la reutilización, debido a que los factores que se identifican en un proyecto son muy específicos como para adecuarlos a otro.

Existen varios métodos para crear modelos de calidad a medida, entre las que se destacan: GQM (Goal-Question-Metric) y la del estándar IEEE 1061.

2.5.1.3. MODELOS DE CALIDAD MIXTOS

Características:

- Combina ventajas de los modelos anteriores.
- Tienen como objetivo identificar factores de calidad abstractos.

- Factores de calidad reutilizables en proyectos que tienen el mismo entorno de desarrollo.
- Posibilidad de refinar factores de calidad y hacerlos operables en proyectos particulares.
- Carecen de información específica de un producto o proyecto, por lo tanto son menos complejos en su estructura.

En este tipo de modelos podemos destacar como propuestas el ADEQUATE (1999), Gilb (1988) y el modelo propuesto por el estándar ISO/IEC 2501n (2005).

2.5.2. PROPIEDADES DE LOS MODELOS DE CALIDAD

Del estudio de las diferentes propuestas de modelos de calidad existentes se desprenden algunas propiedades estructurales importantes:

- **Número de capas:** El número de capas de un modelo de calidad es utilizado como una medida para determinar el nivel de detalle con el que se describe el dominio de software. El número de capas define la siguiente relación proporcional: a más capas o niveles, mayor descomposición y por lo tanto una descripción más detallada del tipo de componente a evaluar. Los modelos a medida tienen una estructura jerárquica con más niveles que los modelos fijos.
- **Tipos de elementos del modelo:** Todos los tipos de modelos incluyen elementos de alto y bajo nivel, los mismos que ayudan a clasificar factores de calidad, describir y evaluar características observables de los componentes. Los términos para referirse a estos tipos de elementos son: factor, atributo, característica, etc.
- **Propósito del modelo:** Las dimensiones que se debe considerar al construir modelos de calidad son las siguientes: específico/general y

reutilizable/descartable. Los modelos generales carecen de información específica de un producto o proyecto, por lo tanto son menos complejos en su estructura (menor número de capas y de elementos de calidad), y son usualmente utilizados como modelos fijos. Por su parte los modelos específicos son construidos a la medida para un producto o proceso dado, debido a esto poseen una mayor cantidad de información disponible, por lo que su estructura final suele ser más compleja.

- Separación entre elementos internos y externos: Los factores externos son todos aquellos factores que pueden ser directamente percibidos por los usuarios (usualmente relacionados a la funcionalidad y usabilidad), mientras que los factores internos hacen referencia a las características constructivas de los componentes que por lo general son accesibles y controlables sólo por sus fabricantes.
- Relaciones entre factores de calidad: Además de la descomposición jerárquica, los modelos incluyen relaciones que involucran a los factores de calidad. Las posibles relaciones son las siguientes:
 - Solapamiento: un factor de calidad participa en la descomposición jerárquica de varios otros niveles superiores. Cabe citar que dicho factor puede evaluarse con métricas diferentes para cada uno de los factores que descompone.
 - Transversalidad: es una relación de solapamiento donde no sólo cambia la métrica, sino también la definición. Este es el caso de las treinta y un subcaracterísticas de cumplimiento asociadas a cada una de las características incluidas en el modelo de calidad del estándar ISO/IEC 2501n.
 - Dependencia: un factor de calidad se relaciona con otros factores, generalmente del mismo nivel. El número de dependencias puede

llegar a ser muy elevado, aunque como señala Egyed y Grübacher, muchas de ellas pueden no ser relevantes.

- Relación de las métricas con los factores de calidad: Todas las propuestas de modelos de calidad existentes incluyen métricas asociadas al menos al nivel más detallado de descomposición, aunque en algunos casos (estándar IEEE 1061), requieren que las métricas sean también aplicadas a los niveles más altos o abstractos de la jerarquía. En el caso del estándar ISO/IEC 9126, las partes 2, 3 y 4, incluyen conjuntos complejos de atributos y métricas explícitamente concebidos para su uso en modelos construidos en base a este estándar.

2.6. MÉTODO IQMC

Fuente: (Coral Calero, 2010)

2.6.1. INTRODUCCIÓN AL MÉTODO IQMC

El método IQMC proporciona un conjunto de guías y técnicas para la identificación de los factores de calidad apropiados que deben ser incluidos en un modelo de calidad que permita analizar la calidad de componentes pertenecientes a un cierto dominio de software. Para empezar a construir el modelo, IQMC se basa en el catálogo de factores del estándar ISO/IEC 25000 descrito anteriormente. Este método adopta un enfoque de modelo mixto.

El desarrollo de IQMC consiste de siete pasos, que pueden ser simultaneados y/o iterados si se considera oportuno. En el primer paso, el ámbito de calidad es explorado en profundidad y en los pasos restantes se establece la construcción del modelo partiendo de las características de calidad y la descomposición en subcaracterísticas del catálogo ISO/IEC 2501n.

2.6.2. ETAPAS O PASOS DEL MODELO IQMC

Tabla 4
Etapas del Método IQMC

No.	Pasos del Método IQMC
0	Estudio del ámbito del Software
1	Determinación de características de calidad
2	Refinamiento de la jerarquía de subcaracterísticas
3	Refinamiento de subcaracterísticas en atributos
4	Refinamiento de atributos derivados en básicos
5	Establecimiento de relaciones entre factores de calidad
6	Determinación de Métricas para los atributos

Fuente: (Coral Calero, 2010)

2.6.2.1. PASO 0: ESTUDIO DEL ÁMBITO DEL SOFTWARE

Este paso consiste en realizar un estudio del ámbito al cual pertenecen los componentes software para los que se quiere evaluar la calidad. Es un paso opcional que puede evitarse en caso de poseer el conocimiento suficiente, de ahí su numeración.

2.6.2.2. PASO 1: DETERMINACIÓN DE CARACTERÍSTICAS DE CALIDAD

Teniendo en cuenta que se parte del catálogo ISO/IEC 25010, el añadido de características no será muy habitual y lo que puede pasar es que alguna de las existentes deba reformularse ligeramente para adaptarla al dominio de interés, o eliminarse en el caso de subcaracterísticas no técnicas.

2.6.2.3. PASO 2: REFINAMIENTO DE LA JERARQUÍA DE SUBCARACTERÍSTICAS

Se descomponen las subcaracterísticas del más bajo nivel de abstracción formando jerarquías de subcaracterísticas. En lo que se refiere a las subcaracterísticas técnicas, al igual que en el paso anterior, el añadido de subcaracterísticas no será muy habitual, excepto en el caso de la descomposición de la subcaracterística “Adecuación” perteneciente a la característica “Funcionalidad”, pues como se ha comentado anteriormente, esta subcaracterística depende del dominio concreto para el cual se construye el modelo. En lo que se refiere a las subcaracterísticas no técnicas, lo que se realizará es una depuración de las subcaracterísticas que no interesen para el proyecto en cuestión (notemos que en el caso de los factores no-técnicos el énfasis es en el proyecto y no en el dominio, dada su naturaleza independiente del dominio, más acusada incluso que en el caso de los factores no-funcionales).

2.6.2.4. PASO 3: REFINAMIENTO DE SUBCARACTERÍSTICAS EN ATRIBUTOS

Este refinamiento tiene como objetivo llegar a tener descompuestas las subcaracterísticas en atributos medibles ya sea de forma directa o indirecta a partir del valor de otros atributos básicos.

2.6.2.5. PASO 4: REFINAMIENTO DE ATRIBUTOS DERIVADOS EN BÁSICOS

Se descomponen los atributos complejos (derivados) hasta obtener atributos básicos, los cuales pueden ser medidos de forma directa.

2.6.2.6. PASO 5: ESTABLECIMIENTO DE RELACIONES ENTRE FACTORES DE CALIDAD

Se establecen las relaciones entre factores de calidad que permiten conocer las dependencias entre los distintos factores de calidad del modelo.

2.6.2.7. PASO 6: DETERMINACIÓN DE MÉTRICAS PARA LOS ATRIBUTOS

Se determinan las métricas para los atributos identificados.

2.7. METODOLOGÍA UWE

Fuente: (Ludwig, 2009)

2.7.1. INTRODUCCIÓN

UWE es un método de ingeniería de software para el desarrollo de aplicaciones web basado en UML. Cualquier tipo de diagrama UML puede ser usado, porque UWE es una extensión de UML.

Dicha metodología está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas, de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Otras características relevantes del proceso y método de autoría son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

2.7.2. ETAPAS

2.7.2.1. ETAPA 1: ANÁLISIS DE REQUISITOS

En esta etapa se fijan los requisitos funcionales de la aplicación Web para reflejarlos en un modelo de casos de uso.

Siguiendo el Proceso de Desarrollo de Software aplicado a la Ingeniería Web y según la introducción del sitio oficial de UWE se propone los casos de uso necesarios para capturar los requisitos del sistema. En esta técnica se define quiénes son los actores (usuarios) de la aplicación y que funcionalidad cumple en el sistema.

2.7.2.2. ETAPA 2: DISEÑO DE CONTENIDO

El diseño conceptual se basa en el análisis de requisitos del paso anterior. Esto incluye los objetos involucrados entre los usuarios y la aplicación.

El modelo conceptual propone construir un modelo de clases con estos objetos, ignorando los aspectos de navegación: Presentación e Interacción que serán tratados posteriormente.

2.7.2.3. ETAPA 3: DISEÑO DE NAVEGACIÓN

Muestra la forma de navegar en el Sistema. En el proceso de construir el modelo espacial de navegación, las decisiones del diseñador están basadas en el modelo conceptual y los requisitos de la aplicación definidos en el modelo de casos de uso. Esta etapa consta de dos actividades de modelado:

- **Modelo del espacio de navegación:** Se realiza el modelado espacial de navegación especificando qué objetos pueden ser visitados a través de la aplicación Web. En éste modelo se representan las clases de navegación que se obtienen a partir de las clases del modelo conceptual.
- **Modelo de la estructura de navegación:** Se realiza el modelado estructural de navegación, mismo que representa la estructura del sitio web, donde se puede incluir elementos como menú superior, menú lateral, hipervínculos al modelo espacial de navegación, dichos elementos de navegación se obtienen a partir de las relaciones entre clases del modelo conceptual.

2.7.2.4. ETAPA 4: DISEÑO DE PRESENTACIÓN

El diseño de presentación soporta la construcción de un modelo basado en el modelo de estructura de navegación y la información adicional recogida durante el análisis de requisitos. El modelo de presentación consiste en mostrar el contenido y la estructura de nodos. Se propone la construcción de bocetos, storyboards y un modelo de flujo de presentación.

Como primer paso, el diseñador Web propone un boceto de cada vista de interfaz del usuario principal, es decir el plan de interfaces abstractas de usuario.

El segundo paso está basado en el modelo del storyboard en el que el diseñador puede decidir si quiere usar la técnica de ventanas múltiples y/o “frames”.

2.7.2.5. ETAPA 5: DISEÑO DE PROCESOS

Este diseño tiene como objetivo representar las acciones de las clases modeladas en la etapa 2.

CAPÍTULO 3: ESTUDIO COMPARATIVO

3.1. INTRODUCCIÓN

El modelo de evaluación a desarrollarse describirá los atributos internos y externos que un framework Front -End de calidad debe poseer, para esto se hará uso de la norma ISO/IEC 20510, siguiendo los lineamientos de la metodología IQMC. El modelo será una base de medición flexible como para evaluar los frameworks jQuery, MooTools y otros que se desarrollen en el mismo entorno (Navegadores Web).

Con lo descrito anteriormente, el modelo ha de representar las diferentes características necesarias para determinar que herramienta (framework) es la más óptima para cubrir las necesidades de desarrollo de software en la empresa SOFYA.

3.2. CONSTRUCCIÓN DEL MODELO DE EVALUACIÓN DE CALIDAD

3.2.1. APLICACIÓN DEL MÉTODO IQMC

A continuación se presenta e implementan los pasos del método IQMC para la construcción del modelo de evaluación de los frameworks:

3.2.1.1. PASO 0: ESTUDIO DEL ÁMBITO DEL SOFTWARE

Es necesario entender que un framework es una abstracción de código común que provee funcionalidades genéricas que pueden ser utilizadas para desarrollar aplicaciones de manera rápida, fácil, modular y sencilla ahorrando tiempo y esfuerzo, entonces, un framework es concreto y también incompleto. Concreto porque es desde un punto de vista simple, un conjunto de componentes; incompleto, porque por sí mismo no puede ser utilizado, ya que

guían a la solución de problemas de programación recurrentes, pero por lo general, no son la solución específica completa (Tavárez, 2009).

Los componentes, funcionalidades y servicios fundamentales que un Framework Front-End debe proporcionar, son los siguientes:

- **Compatibilidad:** Todos los frameworks deben tener componentes que puedan ser interpretados por los navegadores más usados en la actualidad (Internet Explorer, Google Chrome, Mozilla Firefox, Safari y Opera).
- **Comunicación Asíncrona (Ajax):** Usando esta técnica de programación, es fácil para el usuario (desarrollador) usar XMLHttpRequest para gestionar y manipular datos, sin la necesidad de recargar las páginas de un sitio web, aumentando la interactividad y experiencia del usuario.
- **Manipulación y Recorrido del DOM:** Las páginas web creadas sólo con HTML, carecen de dinamismo, por lo tanto no brindan interacción al usuario final. JavaScript es un lenguaje dinámico que posee métodos para acceder, recorrer y manipular objetos DOM.

A medida que las aplicaciones crecen, el código JavaScript se hace difícil de mantener, entorpeciendo la reutilización y la manipulación de elementos de páginas web, debido a esto, los frameworks son creados para aportar métodos que simplifican líneas de código, brindando al usuario la facilidad de agregar, editar, cambiar y eliminar elementos en una página HTML de manera simple y dinámica.

- **Validación de Formularios:** Los frameworks Front-End permiten de una manera relativamente fácil validar campos dentro de uno o varios formularios. Esto, desde el punto de vista del desarrollador, simplifica y reduce el código para procesar dichos formularios, ya que los datos llegan previamente validados, reduciendo los errores de tipos de datos.

- **Efectos visuales:** La idea de emplear efectos visuales en una aplicación web, es que la experiencia de navegación del usuario final sea fácil, amigable e intuitiva. El aporte de los frameworks se destaca en la facilidad que brindan a los desarrolladores para implementar efectos y animaciones, cuanto menos se demore, mayor es el aporte de la herramienta.
- **Manejo de Eventos:** Los eventos son uno de los elementos más importantes en el desarrollo de aplicaciones web enriquecidas (interactivas) del lado del cliente, ya que sirven para realizar acciones en la página a medida que el usuario realiza determinadas acciones con los distintos dispositivos periféricos de entrada. Los eventos son la base para crear la interacción con el usuario.

La importancia de los frameworks en este tema, es la implementación de eventos adicionales a los definidos en JavaScript, así como también métodos que facilitan su manipulación.

- **Soporte:** El equipo o comunidad responsable de un proyecto (framework) es la encargada de dar mantenimiento, publicar nuevas actualizaciones, corregir errores y proporcionar ayuda a través de foros u otros medios. Si el software no tiene un equipo organizado, responsable y comprometido que garantice calidad en un producto, no es recomendable para una empresa invertir recursos para el aprendizaje del mismo.
- **Documentación:** La documentación de los frameworks o cualquier software en general, es importante para identificar fácilmente los aspectos y características que forman parte de los mismos.

Una documentación legible, organizada y actualizada proporciona identidad y popularidad a un proyecto, de manera que los usuarios del mismo podrán identificar fácilmente las ventajas, desventajas, características, funcionalidades, costos y beneficios que impliquen el desarrollo del proyecto.

3.2.1.2. PASO 1: DETERMINACIÓN DE CARACTERÍSTICAS DE CALIDAD

Tabla 5

Características de Calidad de un Framework JavaScript

Características de Calidad		
No.	Nombre	Descripción
1	Funcionalidad	En un framework Front-End, representa las funcionalidades que este brinda al usuario para desarrollar aplicaciones web amigables e interactivas.
2	Rendimiento	Representa los recursos como el tiempo y el número de funciones que un framework utiliza bajo ciertas condiciones.
3	Compatibilidad	Es la capacidad que un framework tiene para llevar a cabo sus funciones sin importar el navegador en el que se ejecute o la coexistencia con otras herramientas.
4	Usabilidad	Este factor evalúa la facilidad con que los usuarios pueden utilizar los diferentes métodos que posee un framework con el fin de alcanzar un objetivo concreto.
5	Fiabilidad	Permite determinar la probabilidad de que un framework desarrolle una cierta función, durante un tiempo establecido y bajo un conjunto de condiciones definidas.
6	Seguridad	Evalúa si un framework posee componentes para la protección de datos en el desarrollo de aplicaciones.

Continúa →

7	Mantenibilidad	Permite determinar la capacidad que tiene un framework para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.
8	Portabilidad	Determina la facilidad que posee un framework para ser instalado, adaptado o reemplazado de forma efectiva y eficiente en diferentes navegadores web.

3.2.1.3. PASO 2: REFINAMIENTO DE LA JERARQUÍA DE SUBCARACTERÍSTICAS DE CALIDAD

Tabla 6

Subcaracterísticas de Funcionalidad

Característica 1		Funcionalidad
Subcaracterísticas		
No.	Nombre	Descripción
1.1	Integridad	Grado en el cual el conjunto de funcionalidades que posee un framework cubre todas las tareas y objetivos planteados por el usuario.
1.2	Precisión	Capacidad del Framework para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.
1.3	Idoneidad	El Framework posee funcionalidades para la comunicación con el servidor, gestión de efectos y eventos, manipulación y recorrido del DOM, Validación de formularios.

Tabla 7

Subcaracterísticas de Rendimiento

Característica 2		Rendimiento
Subcaracterísticas		
No.	Nombre	Descripción
2.1	Comportamiento en el tiempo	Capacidad de un Framework para proporcionar tiempos, procesamiento e índices de respuesta al realizar una funcionalidad específica bajo ciertas condiciones.
2.2	Utilización de recursos	Determina las cantidades y tipos de recursos que utiliza un framework cuando ejecuta una función bajo condiciones específicas.
2.3	Capacidad de Operación	Determina la capacidad de operación de un Framework durante su utilización.

Tabla 8

Subcaracterísticas de Compatibilidad

Característica 3		Compatibilidad
Subcaracterísticas		
No.	Nombre	Descripción
3.1	Coexistencia	Capacidad de un framework para coexistir con otro en un navegador web común, compartiendo recursos comunes sin detrimento.
3.2	Interoperabilidad	Capacidad de dos o más frameworks para intercambiar información y utilizar la información intercambiada.

Tabla 9

Subcaracterísticas de Usabilidad

Característica 4		Usabilidad
Subcaracterísticas		
No.	Nombre	Descripción
4.1	Inteligibilidad	Capacidad de los proveedores del Framework para permitir al usuario identificar si sus funcionalidades son adecuadas para sus necesidades.
4.2	Facilidad de Aprendizaje	Capacidad de la comunidad del Framework para permitirle al usuario aprender los diferentes atributos, métodos y componentes que posee.
4.3	Operatividad	Capacidad del Framework que permite al usuario operarlo y controlarlo con facilidad.
4.4	Protección de errores de usuario	Protección de errores de usuario: Capacidad del sistema para proteger a los usuarios de hacer errores. Refinamiento: No Aplica Motivo: Los Frameworks Front-End no aportan facilidades para detectar y proteger a los desarrolladores de sus errores
4.5	Estética	Capacidad del Framework para proporcionar componentes con estructuras de código agradables y satisfactorias a la vista del desarrollador.
4.6	Accesibilidad	Accesibilidad: Capacidad del producto que permite que sea utilizado por usuarios con determinadas discapacidades. Refinamiento: No Aplica Motivo: Los Frameworks Front-End no poseen funcionalidades o facilidades de esta categoría.

Tabla 10

Subcaracterísticas de Fiabilidad

Característica 5		Fiabilidad
Subcaracterísticas		
No.	Nombre	Descripción
5.1	Madurez	Capacidad del Framework para evitar fallas como resultado de errores en el producto.
5.2	Disponibilidad	Determina si un Framework tiene facilidades de accesibilidad y uso.
5.3	Tolerancia a fallos	Accesibilidad: Capacidad de un Framework para ejecutarse según lo previsto en presencia de fallos software. Refinamiento: No Aplica Motivo: Los Frameworks Front-End no poseen funcionalidades de esta categoría.
5.4	Capacidad de recuperación	Accesibilidad: Capacidad del Framework para recuperar datos afectados y restablecer el estado deseado de la ejecución en caso de interrupción. Refinamiento: No Aplica Motivo: Los Frameworks Front-End no poseen funcionalidades para realizar las tareas mencionadas.

Tabla 11

Subcaracterísticas de Seguridad

Característica 6		Seguridad
Subcaracterísticas		
No.	Nombre	Descripción
6.1	Confidencialidad	Capacidad del Framework para aportar métodos de protección de datos contra el acceso no autorizado. Continúa →

6.2	Integridad	Determina el aporte de métodos para prevenir accesos o modificaciones a funcionalidades no autorizadas.
6.3	No Repudio	Capacidad del Framework para aportar métodos que en el desarrollo de aplicaciones ayuden a demostrar acciones o eventos que realice un determinado usuario.
6.4	Responsabilidad	Autenticidad: Capacidad de rastrear de forma inequívoca las acciones de una entidad. Refinamiento: No Aplica Motivo: Los Frameworks Front-End no poseen métodos o componentes para solventar esta demanda.
6.5	Autenticidad	Autenticidad: Capacidad de demostrar la identidad de un sujeto. Refinamiento: No Aplica Motivo: Los Frameworks Front-End no registran funciones específicas para la demostración de identidades.

Tabla 12

Subcaracterísticas de Mantenibilidad

Característica 7		Mantenibilidad
Subcaracterísticas		
No.	Nombre	Descripción
7.1	Modularidad	Capacidad de un Framework que permite que un cambio en un componente tenga impactos mínimos en los demás.
7.2	Reusabilidad	Capacidad de un Framework que permite la utilización de métodos o componentes en más de un sistema.

Continúa →

7.3	Capacidad de ser analizado	Facilidad que el Framework brinda al usuario para evaluar el impacto de un determinado cambio sobre la implementación de software.
7.4	Capacidad de ser modificado	Capacidad del Framework que permite al usuario modificar código de forma efectiva y eficiente sin introducir defectos o degradar el desempeño de la implementación.
7.5	Capacidad de ser probado	Facilidad para establecer criterios de prueba en el desarrollo de una aplicación.

Tabla 13

Subcaracterísticas de Portabilidad

Característica 8		Portabilidad
Subcaracterísticas		
No.	Nombre	Descripción
8.1	Adaptabilidad	Capacidad del Framework para ser adaptado de forma efectiva y eficiente a diferentes navegadores web.
8.2	Facilidad de Instalación	Facilidad con la que un Framework se puede instalar y/o desinstalar de forma exitosa en una aplicación Web.
8.3	Capacidad de ser reemplazado	Capacidad de un Framework para ser utilizado en lugar de otro, con el mismo propósito y en el mismo navegador Web.

3.2.1.4. PASO 3: REFINAMIENTO DE SUBCARACTERÍSTICAS EN ATRIBUTOS

Tabla 14

Atributos de Integridad

Característica 1		Funcionalidad
Subcaracterística 1.1		Integridad
Atributos		
No.	Nombre	Descripción
1.1.1	Creación de aplicaciones Web interactivas.	Permite la creación de aplicaciones Web donde el receptor (usuario final), tiene una amplia capacidad de control sobre las mismas.
1.1.2	Creación de aplicaciones Web amigables.	Permite la creación de aplicaciones visualmente atractivas y fáciles de usar.
1.1.3	Creación de aplicaciones Web robustas.	Posee funcionalidades para desarrollar aplicaciones web robustas (estables, seguras, alto nivel de procesamiento, etc.).
1.1.4	Creación de aplicaciones Web orientadas a móviles.	Permite la creación de interfaces diseñadas para que las aplicaciones web puedan ser accedidas desde un dispositivo móvil (Smartphone o Tablet).

Tabla 15

Atributos de Precisión

Característica 1		Funcionalidad
Subcaracterística 1.2		Precisión
Atributos		
No.	Nombre	Descripción
1.2.1	Efectividad en la creación de efectos y animaciones.	Posee métodos para monitorear y verificar que los resultados de los efectos y animaciones de una aplicación web sean correctos.
1.2.2	Efectividad en la manipulación del DOM.	Proporciona los suficientes mecanismos para manipular el DOM de manera efectiva.

Tabla 16

Atributos de Idoneidad

Característica 1		Funcionalidad
Subcaracterística 1.3		Idoneidad
Atributos		
No.	Nombre	Descripción
1.3.1	Método Ajax.	Posee el mecanismo Ajax para realizar una comunicación Cliente – Servidor sin recargar una página web.
1.3.2	Métodos para manipular el DOM.	Proporciona funcionalidades para gestionar los elementos de un documento HTML y XHTML.
1.3.3	Elementos interactivos.	Proporciona un conjunto de elementos interactivos configurables, listos para ser utilizados en páginas web.

Continúa →

1.3.4	Validación de formularios.	Posee librerías, componentes y funcionalidades específicas para la validación de formularios.
1.3.5	Efectos visuales.	Posee un conjunto establecido de efectos y la posibilidad de crear nuevos, con propiedades personalizadas por el desarrollador.
1.3.6	Gestión de Tablas o Grids.	Posee librerías y componentes que ayudan a gestionar datos en una tabla.
1.3.7	Pruebas unitarias.	Posee librerías, componentes y funcionalidades para realizar pruebas unitarias.

Tabla 17

Atributos de Comportamiento en el Tiempo

Característica 2		Rendimiento
Subcaracterística 2.1		Comportamiento en el Tiempo
Atributos		
No.	Nombre	Descripción
2.1.1	Tiempo de espera en una comunicación Cliente - Servidor.	Posee métodos para configurar el tiempo de espera de una solicitud realizada desde el cliente hacia el servidor .
2.1.2	Comunicación Síncrona y Asíncrona.	Gestiona el tipo de petición (Síncrona o Asíncrona) que se realiza desde el cliente hasta el servidor, modificando el tiempo de respuesta en una aplicación.
2.1.3	Configuración del tiempo en animaciones y efectos.	Permite configurar la duración de una animación o un efecto.

Tabla 18

Atributos de Rendimiento

Característica 2		Rendimiento
Subcaracterística 2.2		Utilización de Recursos
Atributos		
No.	Nombre	Descripción
2.2.1	Número de librerías	Cantidad de librerías que un framework necesita para para satisfacer las necesidades de un desarrollador (a)

Nota: (a) Ver atributos de la subcaracterística 1.1

Tabla 19

Atributos de Capacidad de Operación

Característica 2		Rendimiento
Subcaracterística 2.3		Capacidad de operación
Atributos		
No.	Nombre	Descripción
2.3.1	Internet Explorer	Capacidad de operación en Internet Explorer.
2.3.2	Safari	Capacidad de operación en Safari.
2.3.3	Mozilla Firefox	Capacidad de operación en Mozilla Firefox.
2.3.4	Google Chrome	Capacidad de operación en Google Chrome.
2.3.5	Opera	Capacidad de operación en Opera.

Tabla 20

Atributos de Coexistencia

Característica 3		Compatibilidad
Subcaracterística 3.1		Coexistencia
Atributos		
No.	Nombre	Descripción
3.1.1	Coexistencia con HTML5	Capacidad del framework para coexistir con HTML5
3.1.2	Coexistencia con XHTML	Capacidad del framework para coexistir con XHTML
3.1.3	Coexistencia con CSS3	Capacidad del framework para coexistir con CSS3
3.1.4	Coexistencia con Flash	Capacidad del framework para coexistir con Flash
3.1.5	Coexistencia con VBscript	Capacidad del framework para coexistir con VBscript

Tabla 21

Atributos de Compatibilidad

Característica 3		Compatibilidad
Subcaracterística 3.2		Interoperabilidad
Atributos		
No.	Nombre	Descripción
3.2.1	Intercambio de información con PHP	Capacidad del framework para intercambiar información con PHP
3.2.2	Intercambio de información con Java EE	Capacidad del framework para intercambiar información con Java EE
3.2.3	Intercambio de información con C# (ASP.net)	Capacidad del framework para intercambiar información con C#

Tabla 22

Atributos de Inteligibilidad

Característica 4		Usabilidad
Subcaracterística 4.1		Inteligibilidad
Atributos		
No.	Nombre	Descripción
4.1.1	Principios bien definidos	Claridad en la definición del framework expuesta en el sitio web y de las funcionalidades integradas en el mismo.

Tabla 23

Atributos de Facilidad de Aprendizaje

Característica 4		Usabilidad
Subcaracterística 4.2		Facilidad de Aprendizaje
Atributos		
No.	Nombre	Descripción
4.2.1	Manual técnico	Disponibilidad de manuales técnicos.
4.2.2	Foros	Disponibilidad de foros para el soporte a usuarios.
4.2.3	Tutoriales	Disponibilidad de tutoriales oficiales y no oficiales.
4.2.4	Cursos on-line	Disponibilidad de cursos on-line oficiales y no oficiales.
4.1.5	Estudios e investigaciones	Se han realizado publicaciones técnicas de las capacidades de la herramienta.

Tabla 24

Atributos de Operatividad

Característica 4		Usabilidad
Subcaracterística 4.3		Operatividad
Atributos		
No.	Nombre	Descripción
4.3.1	POO. basada en Clases.	Se basa en la Programación Orientada a Objetos basada en Clases para el desarrollo de aplicaciones Web.
4.3.2	Reconocimiento de entorno.	Posee funcionalidades que identifican y advierten si un navegador no soporta la versión del framework.

Nota. Según la encuesta realizada en el proyecto (Ver ANEXO C), los desarrolladores de la empresa SOFYA están más familiarizados con la POO basada en clases que con la POO basada en prototipos.

Tabla 25

Atributos de Protección de Errores de Usuario

Característica 4		Usabilidad
Subcaracterística 4.4		Protección de errores de usuario (No Aplica)
Atributos (No Aplica)		

Tabla 26

Atributos de Estética

Característica 4		Usabilidad
Subcaracterística 4.5		Estética
Atributos		
No.	Nombre	Descripción
4.5.1	Sintaxis	La sintaxis de las funcionalidades del Framework, aseguran la completitud y evitar la ambigüedad del código fuente.
4.5.2	Semántica	La semántica que poseen las funcionalidades del Framework aseguran que todas las implementaciones se comporten igual.

Tabla 27

Atributos de Accesibilidad

Característica 4		Usabilidad
Subcaracterística 4.6		Accesibilidad (No Aplica)
Atributos (No Aplica)		

Tabla 28

Atributos de Madurez

Característica 5		Fiabilidad
Subcaracterística 5.1		Madurez
Atributos		
No.	Nombre	Descripción
5.1.1	Tiempo en el mercado	Tiempo que tiene el framework en el mercado y su historial.
		Continúa →

5.1.2	Actualizaciones disponibles	Disponibilidad de actualizaciones al año del producto.
5.1.3	Posee una base del conocimiento software	Mantiene una base del conocimiento de los errores detectados y las soluciones óptimas.

Tabla 29

Atributos de Disponibilidad

Característica 5		Fiabilidad
Subcaracterística 5.2		Disponibilidad
Atributos		
No.	Nombre	Descripción
5.2.1	Descarga del Framework	Posibilidad de descargar el framework para usarlo de forma local.
5.2.2	Referenciar el Framework	Posibilidad de referenciar el framework para usarlo de forma remota.

Tabla 30

Atributos de Tolerancia a Fallos

Característica 5		Fiabilidad
Subcaracterística 5.3		Tolerancia a fallos (No Aplica)
Atributos (No Aplica)		

Tabla 31

Atributos de Capacidad de Recuperación

Característica 5	Fiabilidad
Subcaracterística 5.4	Capacidad de recuperación (No Aplica)
Atributos (No Aplica)	

Tabla 32

Atributos de Confidencialidad

Característica 6	Seguridad	
Subcaracterística 6.1	Confidencialidad	
Atributos		
No.	Nombre	Descripción
6.1.1	Seguridad en transferencia de datos	Implementación de métodos de seguridad para la transferencia de datos Cliente - Servidor.
6.1.2	Tratamiento de contraseñas	Posee componentes que ayudan a la creación de contraseñas seguras.
6.1.3	Encriptación de datos	Posee componentes para la encriptación de datos.

Tabla 33

Atributos de Integridad

Característica 6		Seguridad
Subcaracterística 6.2		Integridad
Atributos		
No.	Nombre	Descripción
6.2.1	Ocultar Funcionalidades	Posibilidad de crear aplicaciones donde se pueda ocultar funcionalidades.
6.2.2	Bloquear Funcionalidades	Posibilidad de crear aplicaciones donde se pueda bloquear funcionalidades.
6.2.3	Restricción de accesos no autorizados	Posee componentes para prevenir accesos no autorizados.

Tabla 34

Atributos de No Repudio

Característica 6		Seguridad
Subcaracterística 6.3		No repudio
Atributos		
No.	Nombre	Descripción
6.3.1	Eventos en formularios	Aporta funcionalidades que ayudan a detectar eventos en determinados formularios.
6.3.2	Eventos de teclado	Aporta funcionalidades para gestionar los eventos del teclado.
6.3.3	Eventos de mouse	Aporta funcionalidades para gestionar los eventos del mouse.

Tabla 35
Atributos de Responsabilidad

Característica 6	Seguridad
Subcaracterística 6.4	Responsabilidad(No Aplica)
Atributos(No Aplica)	

Tabla 36
Atributos de Autenticidad

Característica 6	Seguridad
Subcaracterística 6.5	Autenticidad (No Aplica)
Atributos(No Aplica)	

Tabla 37
Atributos de Modularidad

Característica 7	Mantenibilidad	
Subcaracterística 7.1	Modularidad	
Atributos		
No.	Nombre	Descripción
7.1.1	Framework formado por "widgets" o componentes.	El framework en su mayoría está creado por un conjunto de "widgets" o componentes independientes.
7.1.2	Métodos que se ejecutan en segundo plano.	El framework posee métodos que en una aplicación se cargan en segundo plano, sin interferir con la visualización ni el comportamiento de una página web.

Tabla 38

Atributos de Reusabilidad

Característica 7		Mantenibilidad
Subcaracterística 7.2		Reusabilidad
Atributos		
No.	Nombre	Descripción
7.2.1	Creación de componentes reusables	El framework permite crear componentes parametrizables y reusables.
7.2.2	Anidación	El framework permite anidar componentes para evitar la ambigüedad.

Tabla 39

Atributos de Capacidad de ser Analizado

Característica 7		Mantenibilidad
Subcaracterística 7.3		Capacidad de ser analizado
Atributos		
No.	Nombre	Descripción
7.3.1	Uso de Clases	El framework permite trabajar con clases, agrupando en éstas datos y procedimientos que las manipulan, formando una estructura de información que facilita su entendimiento.

Fuente: (Rivera, 2008)

Tabla 40

Atributos de Capacidad de ser Modificado

Característica 7		Mantenibilidad
Subcaracterística 7.4		Capacidad de ser modificado
Atributos		
No.	Nombre	Descripción
7.4.1	Código Abierto	Los creadores del framework, facilitan el código fuente para modificarlo con libre albedrío.
7.4.2	Actualización de funcionalidades	Permite extender el alcance de funcionalidades nativas del framework.

Tabla 41

Atributos de Capacidad de ser Probado

Característica 7		Mantenibilidad
Subcaracterística 7.5		Capacidad de ser probado
Atributos		
No.	Nombre	Descripción
7.5.1	Pruebas unitarias	Posee funcionalidades o librerías especializadas para realizar pruebas unitarias.

Tabla 42

Atributos de Adaptabilidad

Característica 8		Portabilidad
Subcaracterística 8.1		Adaptabilidad
Atributos		
No.	Nombre	Descripción
8.1.1	Adaptación de sintaxis	Permite usar las funcionalidades de un framework adaptando su sintaxis a la de otro.
		Continúa →

8.1.2	Funciona en Internet Explorer	Capacidad del Framework para operar en el navegador web Internet Explorer.
8.1.3	Funciona en Safari	Capacidad del Framework para operar en el navegador web Safari.
8.1.4	Funciona en Mozilla Firefox	Capacidad del Framework para operar en el navegador web Mozilla Firefox.
8.1.5	Funciona en Google Chrome	Capacidad del Framework para operar en el navegador web Google Chrome.
8.1.6	Funciona en Opera	Capacidad del Framework para operar en el navegador web Opera.
8.1.7	Funciona en Back-End y Front-End.	Capacidad del Framework para adaptar funcionalidades en Back-End y Front-End.
8.1.8	Funciona en navegadores web para móviles.	Posee funcionalidades para crear interfaces orientadas a los Smartphone.

Tabla 43

Atributos de Facilidad de Instalación

Característica 8		Portabilidad
Subcaracterística 8.2		Facilidad de instalación
Atributos		
No.	Nombre	Descripción
8.2.1	Manuales de instalación	Provee manuales de instalación.
8.2.2	Ayuda en línea	Los proveedores o los usuarios del framework, a través de foros o archivos multimedia proveen ayudas para la instalación.
8.2.3	Tiempo de instalación	Tiempo que toma a los usuarios instalar el framework.

Tabla 44

Atributos de Capacidad de ser Reemplazado

Característica 8		Portabilidad
Subcaracterística 8.3		Capacidad de ser reemplazado
Atributos		
No.	Nombre	Descripción
8.3.1	Reemplazo de versiones (con compresión y sin compresión)	Facilidad para cambiar un framework descomprimido por uno comprimido en una aplicación terminada. Framework versión comprimida: Disminuye el peso de los archivos en una aplicación. Código fuente ofuscado. Framework versión descomprimida: Aumenta el peso de los archivos en una aplicación. Código fuente sin ofuscar.
8.3.2	Cambio de versiones (actualización de Framework)	Facilidad para reemplazar archivos de un Framework antiguo por uno nuevo.
8.3.3	Reemplazo de Framework	Facilidad para reemplazar un Framework por otro de diferente proveedor.

3.2.1.5. PASO 4: REFINAMIENTO DE ATRIBUTOS DERIVADOS EN BÁSICOS

Tabla 45

Atributos básicos de Efectividad en la creación de efectos y animaciones

Característica 1		Funcionalidad
Subcaracterística 1.2		Precisión
Atributo derivado 1.2.1		Efectividad en la creación de efectos y animaciones
Atributos Básicos		
No.	Nombre	Descripción
1.2.1.1	Configuración de fps (Frames por segundo).	Permite crear animaciones con el nivel de fluidez requerido.
1.2.1.2	Configuración de unit (Unidad de medida).	Permite definir la unidad de medida para la ubicación y propagación apropiada de animaciones y efectos.
1.2.1.3	Configuración de tiempo.	Permite crear efectos y animaciones con tiempos de ejecución configurables.
1.2.1.4	Adaptación automática.	Permite identificar de forma automática las dimensiones de un entorno específico, para crear elementos, efectos y animaciones que se ajusten a este.

Tabla 46

Atributos Básicos de Efectividad en la Manipulación del DOM

Característica 1		Funcionalidad
Subcaracterística 1.2		Precisión
Atributo derivado 1.2.2		Efectividad en la manipulación del DOM
Atributos Básicos		
No.	Nombre	Descripción
1.2.2.1	Selectores "id" y "class" .	Contienen métodos para seleccionar elementos del DOM a través de las propiedades "id" y "class".
1.2.2.2	Selectores CSS.	Provee métodos para seleccionar elementos del DOM de manera similar a la que se usa en las hojas de estilo.
1.2.2.3	Selectores por tipo de Elemento.	Permite seleccionar elementos del DOM haciendo uso del tipo. Ejemplo: "table", "div", "ul", etc.
1.2.2.4	Selectores personalizados.	Provee un conjunto de métodos que integran operadores aritméticos y de comparación para facilitar las selecciones.

Tabla 47

Atributos básicos de Sintaxis

Característica 4		Usabilidad
Subcaracterística 4.5		Estética
Atributo derivado 4.5.1		Sintaxis
Atributos Básicos		
No.	Nombre	Descripción
4.5.1.1	Claridad	Claridad en la sintaxis de las funcionalidades del Framework.
4.5.1.2	Simplicidad	Simplicidad en la sintaxis de las funcionalidades del Framework.

Tabla 48

Atributos básicos de Semántica

Característica 4		Usabilidad
Subcaracterística 4.5		Estética
Atributo derivado 4.5.2		Semántica
Atributos Básicos		
No.	Nombre	Descripción
4.5.2.3	Unidad de conceptos	Métodos que representan un número mínimo de conceptos unificados, cuyas reglas sean sencillas y regulares.

Tabla 49

Atributos básicos de Descarga del Framework

Característica 5		Usabilidad
Subcaracterística 5.2		Disponibilidad
Atributo derivado 5.2.1		Descarga del Framework
Atributos Básicos		
No.	Nombre	Descripción
5.2.1.1	Descarga con compresión	Permite descargar el framework con compresión para usar en aplicaciones en producción.
5.2.1.2	Descarga sin compresión	Permite descargar el framework sin compresión para usar en aplicaciones en desarrollo.
5.2.1.3	Descarga desde el sitio web oficial	Permite descargar el framework con o sin compresión desde el sitio web oficial.
5.2.1.4	Descarga desde sitios web no oficiales	Permite descargar el framework con o sin compresión desde sitios web no oficiales.

Tabla 50

Atributos básicos de Referenciar el Framework

Característica 5		Usabilidad
Subcaracterística 5.2		Disponibilidad
Atributo derivado 5.2.2		Referenciar el Framework
Atributos Básicos		
No.	Nombre	Descripción
5.2.2.1	Referencia a través del sitio web oficial	Permite referenciar el framework a través del sitio web oficial.
5.2.2.2	Referencia a través de Google	Permite referenciar el framework a través de Google.
5.2.2.3	Referencia a través de sitios web no oficiales	Permite referenciar el framework a través de sitios web no oficiales.

3.2.1.6. PASO 5: ESTABLECIMIENTO DE RELACIONES ENTRE FACTORES DE CALIDAD

Para el desarrollo del modelo de calidad se mantienen las relaciones jerárquicas de profundidad que propone la norma ISO/IEC 25000.

La Figura 8 indica el tipo de relación que se usará en el desarrollo del modelo de calidad:

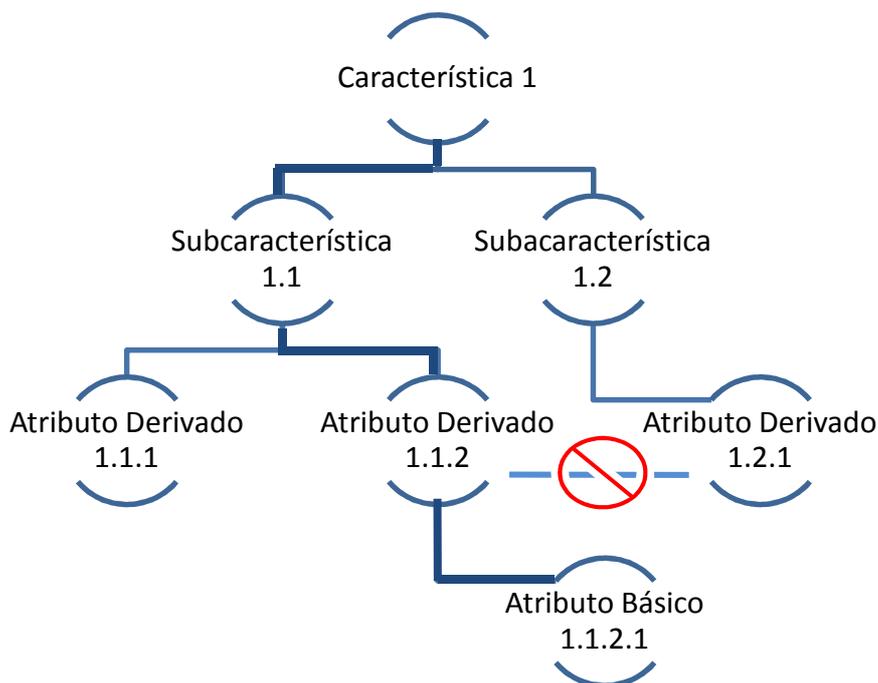


Figura 8 Relaciones Entre Factores de Calidad

El objetivo de mantener este tipo de relación es crear una estructura de factores organizados por grupos o categorías que se descomponen por criterios afines al tipo de software a evaluar y se miden de manera independiente. Un modelo con estas características evita la dependencia entre atributos, fomentando la flexibilidad y reutilización.

3.2.1.7. PASO 6: DETERMINACIÓN DE MÉTRICAS PARA LOS ATRIBUTOS

Para evaluar las características de calidad (norma ISO/IEC 25010) de un framework y seleccionar el más idóneo para el desarrollo de aplicaciones en la empresa SOFYA, se hará uso de métricas objetivas (medidas directas), que determinarán el nivel de cumplimiento que cada software posee frente a los atributos de calidad planteados en los pasos 3 y 4 del método IQMC.

A continuación se detallan las métricas que se usarán en el modelo de evaluación:

- Evaluación del Cumplimiento de las Características:

Tabla 51

Métrica de Cumplimiento

No.	Cumplimiento	Equivalente contable
1	Si	1
2	No	0

- Evaluación del Cumplimiento de las Características por Rango:

Tabla 52

Métrica de Cumplimiento por Rangos

No.	Intervalos	Equivalente contable	Cumplimiento
1	0	0	Nulo
2	1	0,25	Bajo
3	2	0,50	Medio
4	3	0,75	Alto
5	4	1	Completo

3.2.2. MODELO INTEGRADO

Tabla 53

Modelo de Evaluación de Calidad

■	Característica		
■	Subcaracterística		
■	Atributo derivado		
■	Atributo básico		
1	Funcionalidad		
1.1	Integridad		Métrica
	1.1.1	Creación de aplicaciones Web interactivas	SI = 1, NO = 0
	1.1.2	Creación de aplicaciones Web amigables	SI = 1, NO = 0
	1.1.3	Creación de aplicaciones Web robustas	SI = 1, NO = 0
	1.1.4	Creación de aplicaciones Web orientadas a móviles	SI = 1, NO = 0
1.2	Precisión		
	1.2.1	Efectividad en la creación de efectos y animaciones	Métrica
	1.2.1.1	Configuración de "fps" (Frames por segundo)	SI = 1, NO = 0
	1.2.1.2	Configuración de "unit" (Unidad de medida)	SI = 1, NO = 0
	1.2.1.3	Configuración de tiempo	SI = 1, NO = 0
	1.2.1.4	Adaptación automática	SI = 1, NO = 0
	1.2.2	Efectividad en la manipulación del DOM	Métrica
	1.2.2.1	Selectores "id" y "class"	SI = 1, NO = 0
	1.2.2.2	Selectores CSS	SI = 1, NO = 0
	1.2.2.3	Selectores por tipo de Elemento	SI = 1, NO = 0
	1.2.2.4	Selectores personalizados	SI = 1, NO = 0
1.3	Idoneidad		Métrica
	1.3.1	Método Ajax.	SI = 1, NO = 0
	1.3.2	Métodos para manipular el DOM	SI = 1, NO = 0
	1.3.3	Elementos interactivos	SI = 1, NO = 0
	1.3.4	Validación de formularios	SI = 1, NO = 0
	1.3.5	Efectos visuales	SI = 1, NO = 0
	1.3.6	Gestión de Tablas o Grids	SI = 1, NO = 0
	1.3.7	Pruebas unitarias	SI = 1, NO = 0
2	Rendimiento		
2.1	Comportamiento en el Tiempo		Métrica
	2.1.1	Tiempo de espera en una comunicación Cliente - Servidor	SI = 1, NO = 0
	2.1.2	Comunicación Síncrona y Asíncrona	SI = 1, NO = 0
	2.1.3	Configuración de animaciones y efectos en tiempo de ejecución.	SI = 1, NO = 0
2.2	Utilización de Recursos	Métrica	Evaluación
	2.2.1	Número de librerías	[0:4] A mayor recursos, menor calificación: 1 librería: [4] 2 librerías: [3] 3 librerías: [2] 4 librerías: [1] >4 librerías: [0]

Continúa →

2.3		Capacidad de operación	Métrica	Evaluación
	2.3.1	Capacidad de operación en Internet Explorer	[0:4]	A mayor velocidad de procesamiento, mayor calificación: La escala fue tomada de (MooTools, 2006). <100 ms : [4] 200 ms : [3] 300 ms : [2] 400 ms : [1] >400 ms : [0]
	2.3.2	Capacidad de operación en Safari		
	2.3.3	Capacidad de operación en Mozilla Firefox		
	2.3.4	Capacidad de operación en Google Chrome		
	2.3.5	Capacidad de operación en Opera		
3 Compatibilidad				
3.1 Coexistencia			Métrica	
	3.1.1	Coexistencia con HTML5	SI = 1, NO = 0	
	3.1.2	Coexistencia con XHTML	SI = 1, NO = 0	
	3.1.3	Coexistencia con CSS3	SI = 1, NO = 0	
	3.1.4	Coexistencia con Flash	SI = 1, NO = 0	
	3.1.5	Coexistencia con VBscript	SI = 1, NO = 0	
3.2 Interoperabilidad			Métrica	
	3.2.1	Intercambio de información con PHP	SI = 1, NO = 0	
	3.2.2	Intercambio de información con Java EE	SI = 1, NO = 0	
	3.2.3	Intercambio de información con C# (ASP.net)	SI = 1, NO = 0	
4 Usabilidad				
4.1 Inteligibilidad		Métrica	Evaluación	
	4.1.1	Principios bien definidos	[0:4]	A mayor claridad en las definiciones, mayor calificación. La calificación será en base a los resultados de la encuesta realizada al personal técnico de SOFYA.
4.2 Facilidad de Aprendizaje				
	4.2.1	Manual técnico	[0:4]	A mayor recursos disponibles, mayor calificación. La calificación será en base a los resultados de la encuesta realizada al personal técnico de SOFYA.
	4.2.2	Foros		
	4.2.3	Tutoriales		
	4.2.4	Cursos on-line		
	4.2.5	Estudios e investigaciones		
4.3 Operatividad			Métrica	
	4.3.1	POO. basada en Clases	SI = 1, NO = 0	
	4.3.2	Reconocimiento de entorno	SI = 1, NO = 0	
4.4 Protección de errores de usuario (N/A)				
4.5 Estética				
4.5.1 Sintaxis		Métrica	Evaluación	
	4.5.1.1	Claridad	[0:4]	A mayor claridad, mayor calificación. La calificación será en base a los resultados de la encuesta realizada al personal técnico de SOFYA.
	4.5.1.2	Simplicidad	[0:4]	A mayor simplicidad, mayor calificación. La calificación será en base a los resultados de la encuesta realizada al personal técnico de SOFYA.

Continúa →

	4.5.2	Semántica		Métrica	Evaluación
		4.5.2.1	Unidad de conceptos	[0:4]	A mayor unificación de conceptos con reglas sencillas y regulares, mayor calificación. La calificación será en base a los resultados de la encuesta realizada al personal técnico de SOFYA.
	4.6	Accesibilidad (N/A)			
	5	Fiabilidad			
	5.1	Madurez		Métrica	Evaluación
		5.1.1	Tiempo en el mercado	[0:4]	A mayor tiempo, mayor calificación. 1 año: [0] 2 años: [1] 3 años: [2] 4 años: [3] > 4 años: [4]
		5.1.2	Actualizaciones disponibles	[0:4]	A mayor recursos publicados en el tiempo de vida en el mercado, mayor calificación. 1: [0] 5: [1] 10: [2] 15: [3] > 15: [4]
		5.1.3	Posee una base del conocimiento software	SI = 1, NO = 0	
	5.2	Disponibilidad			
		5.2.1	Descarga del Framework		Métrica
			5.2.1.1	Descarga con compresión	SI = 1, NO = 0
			5.2.1.2	Descarga sin compresión	SI = 1, NO = 0
			5.2.1.3	Descarga desde el sitio web oficial	SI = 1, NO = 0
			5.2.1.4	Descarga desde sitios web no oficiales	SI = 1, NO = 0
		5.2.2	Referenciar el Framework		Métrica
			5.2.2.1	Referencia a través del sitio web oficial	SI = 1, NO = 0
			5.2.2.2	Referencia a través de Google	SI = 1, NO = 0
			5.2.2.3	Referencia a través de sitios web no oficiales	SI = 1, NO = 0
	5.3	Tolerancia a fallos (N/A)			
	5.4	Capacidad de recuperación (N/A)			
	6	Seguridad			
		6.1	Confidencialidad		Métrica
			6.1.1	Seguridad en transferencia de datos	SI = 1, NO = 0
			6.1.2	Tratamiento de contraseñas	SI = 1, NO = 0
			6.1.3	Encriptación de datos	SI = 1, NO = 0
		6.2	Integridad		Métrica
			6.2.1	Ocultar Funcionalidades	SI = 1, NO = 0
			6.2.2	Bloquear Funcionalidades	SI = 1, NO = 0
			6.2.3	Restricción de accesos no autorizados	SI = 1, NO = 0
		6.3	No repudio		Métrica
			6.3.1	Eventos en formularios	SI = 1, NO = 0
			6.3.2	Eventos de teclado	SI = 1, NO = 0
			6.3.3	Eventos de mouse	SI = 1, NO = 0
		6.4	Responsabilidad(N/A)		
		6.5	Autenticidad(N/A)		
		Continúa →			

7	Mantenibilidad		
	7.1	Modularidad	Métrica
	7.1.1	Framework formado por widgets o componentes	SI = 1, NO = 0
	7.1.2	Métodos que se ejecutan en segundo plano	SI = 1, NO = 0
	7.2	Reusabilidad	Métrica
	7.2.1	Creación de componentes reusables	SI = 1, NO = 0
	7.2.2	Anidación	SI = 1, NO = 0
	7.3	Capacidad de ser analizado	Métrica
	7.3.1	Uso de Clases	SI = 1, NO = 0
	7.4	Capacidad de ser modificado	Métrica
	7.4.1	Código Abierto	SI = 1, NO = 0
	7.4.2	Actualización de funcionalidades	SI = 1, NO = 0
	7.5	Capacidad de ser probado	Métrica
	7.5.1	Pruebas unitarias	SI = 1, NO = 0
8	Portabilidad		
	8.1	Adaptabilidad	Métrica
	8.1.1	Adaptación de sintaxis	SI = 1, NO = 0
	8.1.2	Funciona en Internet Explorer	SI = 1, NO = 0
	8.1.3	Funciona en Safari	SI = 1, NO = 0
	8.1.4	Funciona en Mozilla Firefox	SI = 1, NO = 0
	8.1.5	Funciona en Google Chrome	SI = 1, NO = 0
	8.1.6	Funciona en Opera	SI = 1, NO = 0
	8.1.7	Funciona en Back-End y Front-End	SI = 1, NO = 0
	8.1.8	Funciona en navegadores web para móviles	SI = 1, NO = 0
	8.2	Facilidad de instalación	Métrica Evaluación
	8.2.1	Manuales de instalación	SI = 1, NO = 0
	8.2.2	Ayuda en línea	[0:4] A mayor recursos identificados en el transcurso del desarrollo del proyecto, mayor calificación.
	8.2.3	Tiempo de instalación	[0:4] A menor tiempo, mayor calificación. 0 segundos: [4] [1-30] segundos: [3] 1 minuto: [2] 3 minutos: [1] >3 minutos: [0]
	8.3	Capacidad de ser reemplazado	Métrica Evaluación
	8.3.1	Reemplazo de versiones (con compresión y sin compresión)	[0:4]
	8.3.2	Cambio de versiones (actualización de Framework)	A mayor dificultad, menor calificación
	8.3.3	Reemplazo de Framework	

Nota: (N/A) = No Aplica

3.3. APLICACIÓN DEL MODELO

A continuación se aplica el modelo desarrollado anteriormente en los frameworks JavaScript: jQuery y MooTools

Tabla 54

Aplicación del Modelo de Evaluación de Calidad

		Característica		Subcaracterística		Atributo derivado		Atributo básico	
Características, Subcaracterísticas, Atributos derivados y Atributos básicos				Métrica	jQuery	MooTools			
1 Funcionalidad									
1.1 Integridad									
		1.1.1	Creación de aplicaciones Web interactivas	1,0	1	1			
		1.1.2	Creación de aplicaciones Web amigables	1,0	1	1			
		1.1.3	Creación de aplicaciones Web robustas	1,0	1	1			
		1.1.4	Creación de aplicaciones Web orientadas a móviles	1,0	1	1			
1.2 Precisión									
		1.2.1	Efectividad en la creación de efectos y animaciones						
		1.2.1.1	Configuración de "fps" (Frames por segundo)	1,0	0	1			
		1.2.1.2	Configuración de "unit" (Unidad de medida)	1,0	0	1			
		1.2.1.3	Configuración de tiempo	1,0	1	1			
		1.2.1.4	Adaptación automática	1,0	1	1			
		1.2.2	Efectividad en la manipulación del DOM						
		1.2.2.1	Selectores "id" y "class"	1,0	1	1			
		1.2.2.2	Selectores CSS	1,0	1	1			
		1.2.2.3	Selectores por tipo de Elemento	1,0	1	1			
		1.2.2.4	Selectores personalizados	1,0	1	1			
1.3 Idoneidad									
		1.3.1	Método Ajax.	1,0	1	1			
		1.3.2	Métodos para manipular el DOM	1,0	1	1			
		1.3.3	Elementos interactivos	1,0	1	1			
		1.3.4	Validación de formularios	1,0	1	1			
		1.3.5	Efectos visuales	1,0	1	1			
		1.3.6	Gestión de "Grids"	1,0	1	1			
		1.3.7	Pruebas unitarias	1,0	1	0			
2 Rendimiento									
2.1 Comportamiento en el Tiempo									
		2.1.1	Tiempo de espera en una comunicación Cliente - Servidor	1,0	1	0			
		2.1.2	Comunicación Síncrona y Asíncrona	1,0	1	1			
		2.1.3	Configuración del tiempo en animaciones y efectos	1,0	1	1			
2.2 Utilización de Recursos									
		2.2.1	Número de librerías	0:4	1	3			
2.3 Capacidad de operación, Resultados de: (Velichkov, 2007)									
		2.3.1	Capacidad de operación en Internet Explorer	0:4	2	3			
		2.3.2	Capacidad de operación en Safari	0:4	3	4			
		2.3.3	Capacidad de operación en Mozilla Firefox	0:4	3	2			
		2.3.4	Capacidad de operación en Google Chrome	0:4	4	4			
		2.3.5	Capacidad de operación en Opera	0:4	3	2			
3 Compatibilidad									
3.1 Coexistencia									
		3.1.1	Coexistencia con HTML5	1,0	1	1			

Continúa →

	3.1.2	Coexistencia con XHTML	1,0	1	1
	3.1.3	Coexistencia con CSS3	1,0	1	1
	3.1.4	Coexistencia con Flash	1,0	1	0
	3.1.5	Coexistencia con VBscript	1,0	1	1
	3.2	Interoperabilidad			
	3.2.1	Intercambio de información con PHP	1,0	1	1
	3.2.2	Intercambio de información con Java EE	1,0	1	1
	3.2.3	Intercambio de información con C# (ASP.net)	1,0	1	1
4	Usabilidad				
	4.1	Inteligibilidad. (a), Resultados			
	4.1.1	Principios bien definidos	0:4	3	2
	4.2	Facilidad de Aprendizaje. (b), Resultados			
	4.2.1	Manual técnico	0:4	3	2
	4.2.2	Foros	0:4	4	2
	4.2.3	Tutoriales	0:4	4	3
	4.2.4	Cursos on-line	0:4	4	2
	4.2.5	Estudios e investigaciones	0:4	3	1
	4.3	Operatividad			
	4.3.1	POO. basada en Clases.	1,0	0	1
	4.3.2	Reconocimiento de entorno	1,0	1	1
	4.4	Protección de errores de usuario (N/A)			
	4.5	Estética. (c), Resultados			
	4.5.1	Sintaxis			
	4.5.1.1	Claridad	0:4	2	4
	4.5.1.2	Simplicidad	0:4	3	4
	4.5.2	Semántica			
	4.5.2.1	Unidad de conceptos	0:4	3	3
	4.6	Accesibilidad (N/A)			
5	Fiabilidad				
	5.1	Madurez			
	5.1.1	Tiempo en el mercado	0:4	4	3
	5.1.2	Actualizaciones disponibles	0:4	4	1
	5.1.3	Posee una base del conocimiento software	1,0	1	0
	5.2	Disponibilidad			
	5.2.1	Descarga del Framework			
	5.2.1.1	Descarga con compresión	1,0	1	1
	5.2.1.2	Descarga sin compresión	1,0	1	1
	5.2.1.3	Descarga desde el sitio web oficial	1,0	1	1
	5.2.1.4	Descarga desde sitios web no oficiales	1,0	1	1
	5.2.2	Referenciar el Framework			
	5.2.2.1	Referencia a través del sitio web oficial	1,0	1	0
	5.2.2.2	Referencia a través de Google	1,0	1	1
	5.2.2.3	Referencia a través de sitios web no oficiales	1,0	1	0
	5.3	Tolerancia a fallos (N/A)			
	5.4	Capacidad de recuperación (N/A)			
6	Seguridad				
	6.1	Confidencialidad			
	6.1.1	Seguridad en transferencia de datos	1,0	1	1
	6.1.2	Tratamiento de contraseñas	1,0	1	1
	6.1.3	Encriptación de datos	1,0	1	0
	6.2	Integridad			
	6.2.1	Ocultar Funcionalidades	1,0	1	1
	6.2.2	Bloquear Funcionalidades	1,0	1	1
	6.2.3	Restricción de accesos no autorizados	1,0	1	1
	6.3	No repudio			
	6.3.1	Eventos en formularios	1,0	1	1
	6.3.2	Eventos de teclado	1,0	1	1
	6.3.3	Eventos de mouse	1,0	1	1

Continúa →

	6.4	Responsabilidad (N/A)			
	6.5	Autenticidad (N/A)			
7	Mantenibilidad				
	7.1	Modularidad			
	7.1.1	Framework formado por "widgets" o componentes	1,0	0	1
	7.1.2	Métodos que se ejecutan en segundo plano	1,0	1	1
	7.2	Reusabilidad			
	7.2.1	Creación de componentes reusables	1,0	1	1
	7.2.2	Anidación	1,0	0	1
	7.3	Capacidad de ser analizado			
	7.3.1	Uso de Clases	1,0	0	1
	7.4	Capacidad de ser modificado			
	7.4.1	Código Abierto	1,0	1	1
	7.4.2	Actualización de funcionalidades	1,0	1	1
	7.5	Capacidad de ser probado			
	7.5.1	Pruebas unitarias	1,0	1	0
8	Portabilidad				
	8.1	Adaptabilidad			
	8.1.1	Adaptación de sintaxis	1,0	0	1
	8.1.2	Funciona en Internet Explorer	1,0	1	1
	8.1.3	Funciona en Safari	1,0	1	1
	8.1.4	Funciona en Mozilla Firefox	1,0	1	1
	8.1.5	Funciona en Google Chrome	1,0	1	1
	8.1.6	Funciona en Opera	1,0	1	1
	8.1.7	Funciona en Back-End y Front-End	1,0	0	1
	8.1.8	Funciona en navegadores web para móviles	1,0	1	1
	8.2	Facilidad de instalación			
	8.2.1	Manuales de instalación	1,0	1	1
	8.2.2	Ayuda en línea	0:4	4	1
	8.2.3	Tiempo de instalación	0:4	1	1
	8.3	Capacidad de ser reemplazado			
	8.3.1	Reemplazo de versiones (con compresión y sin compresión)	0:4	1	1
	8.3.2	Cambio de versiones (actualización de Framework)	0:4	1	1
	8.3.3	Reemplazo de Framework	0:4	4	2

Nota: (N/A) = No Aplica, (a, b, c) Ver Anexo C

3.4. RESULTADO DEL ANÁLISIS COMPARATIVO

A continuación se describen los resultados del análisis comparativo, el mismo se encuentra organizado en base a las características de calidad de la norma ISO/IEC 25000.

Tabla 55

Resultado del Análisis Comparativo

No.	Características de Calidad	Puntos	%	jQuery		MooTools	
				Puntos	%	Puntos	%
1	Funcionalidad	19	21,59	17	19,32	18	20,45
2	Rendimiento	9	10,23	7	7,95	6,5	7,39
3	Compatibilidad	8	9,09	8	9,09	7	7,95
4	Usabilidad	11	12,50	8,25	9,38	7,75	8,81
5	Fiabilidad	10	11,36	10	11,36	6	6,82
6	Seguridad	9	10,23	9	10,23	8	9,09
7	Mantenibilidad	8	9,09	5	5,68	7	7,95
8	Portabilidad	14	15,91	9,75	11,08	10,75	12,22
TOTAL		88	100,00	74	84,09	71	80,68

La Figura 9 indica los detalles de los resultados de la evaluación de las dos herramientas (jQuery y MooTools), en base a los factores de calidad.

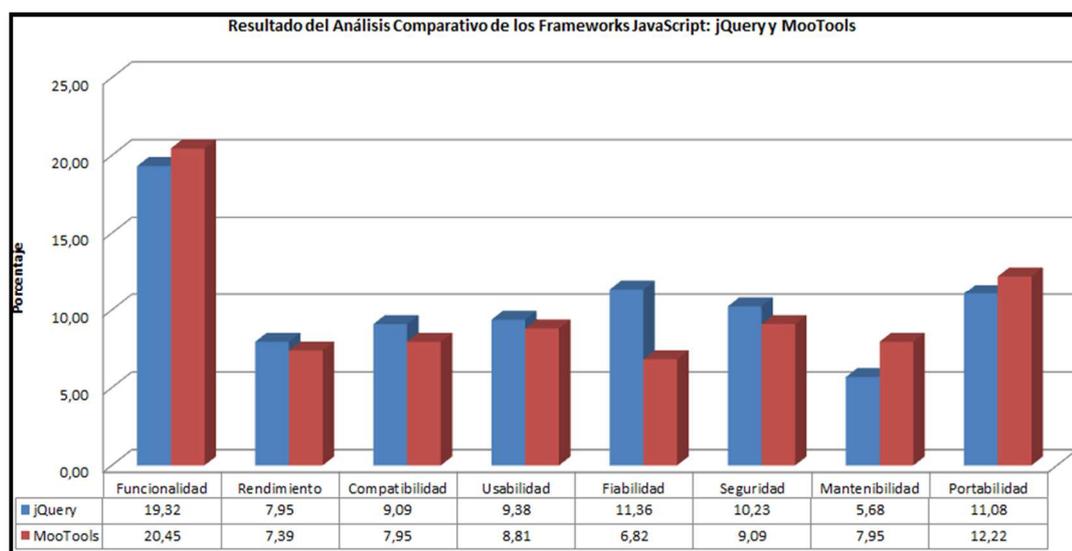


Figura 9 Resultado del Análisis Comparativo de los Frameworks

3.5. INTERPRETACIÓN

Una vez realizada la evaluación de los frameworks JavaScript con el modelo desarrollado, se puede observar que ambas herramientas brindan funcionalidades acorde a las necesidades de la empresa SOFYA. MooTools proporciona un mayor número de funcionalidades para desarrollar aplicaciones con mayor precisión que jQuery, pero el rendimiento que éste framework posee no es muy favorable a la hora de ejecutarse en los navegadores.

jQuery hace énfasis en la creación de componentes que son compatibles con el mayor número de herramientas que se ejecutan en un entorno Web, en cambio MooTools se enfoca en crear funcionalidades que permiten al desarrollador escribir código flexible y potente.

La inmensa comunidad de jQuery ha permitido que la curva de aprendizaje de ésta herramienta sea menos pronunciada que la de MooTools, debido a esto, un gran número de componentes dedicados a la seguridad han sido creados y puestos al servicio de todos los desarrolladores, permitiendo la creación de aplicaciones Web más seguras.

Por otro lado, los creadores de MooTools han considerado la demanda de complejidad en el desarrollo de aplicaciones Web del lado del cliente y para cubrir dicha demanda, han ofertado una herramienta que facilita al desarrollador la mantenibilidad de código.

Ambos frameworks tienen fortalezas y debilidades, pero después de aplicar el modelo de evaluación, se puede llegar a la conclusión de que jQuery supera a MooTools como herramienta para el desarrollo de aplicaciones Web en la empresa SOFYA.

Tomando en cuenta la conclusión anterior, se procede a seleccionar a jQuery como el framework que dará soporte al desarrollo de la aplicación SCPC (Sistema de Comercialización y Producción de Concreto).

CAPÍTULO 4:

DESARROLLO DE LA APLICACIÓN SPCP (SISTEMA DE COMERCIALIZACIÓN Y PRODUCCIÓN DE CONCRETO)

4.1. ANÁLISIS DE REQUISITOS.

En UWE el modelado de requisitos consiste de dos partes:

- Casos de uso de la aplicación.
- Descripción de los casos de uso.

(Ludwig, 2009)

4.1.1. IDENTIFICACIÓN DE LOS ACTORES (USUARIOS):

- Administrador: Es la persona encargada de la gestión de las siguientes entidades: usuario, perfil, cliente, obra, hormigón, elemento-obra, aditivo, carro y chofer. Este usuario también tiene permisos para administrar tareas que realizan los actores citados a continuación.
- Programador de pedidos: Es la persona encargada de registrar, editar, visualizar y eliminar pedidos de hormigón en una agenda. Para registrar un pedido, el usuario recibe a través de una o varias llamadas telefónicas la siguiente información:
 - Fecha de entrega del pedido.
 - Si la descarga de hormigón va a necesitar o no de una Bomba.
 - Hora de inicio.
 - Hora fin.
 - Nombre del cliente que solicita el hormigón.
 - Nombre de la obra del cliente.

- Dirección de la obra.
 - Tipo de Hormigón.
 - Cantidad de volumen.
 - Forma de pago.
 - El elemento que se va a fundir.
 - Aditivo para agregar al hormigón.
 - Crear o no una inspección para la obra del cliente.
 - Observación.
-
- Inspector de obras: Este usuario está encargado de visitar la obra donde se transportará el hormigón y de registrar en el sistema los datos que observó en la visita. Los posibles datos son:
 - Número de inspección.
 - Nombre de la persona encargada del encaje del volumen.
 - Hora de la Inspección.
 - Área de la construcción.
 - Espesor del elemento a fundir.
 - Medidas del tipo de bloque que se usará en la construcción.
-
- Despachador de pedidos: Este usuario se encarga de gestionar los despachos de hormigón una vez que se hayan inspeccionado las obras correspondientes. Las posibles funciones del usuario en el sistema son:
 - Asignar camiones para el transporte de hormigón.
 - Asignar choferes para los camiones.
 - Coordinar el tiempo de salida de los Camiones.
 - Editar guías de remisión.
 - Imprimir guías.
 - Anular guías.

- Visualizar los datos que se imprimirán en una guía.
- Crear cilindros de hormigón para examinarlos en el laboratorio.

4.1.2. CASOS DE USO

En la Figura 10 se detalla la forma en la que un colaborador de la Empresa Hormigonera Equinoccial opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan.

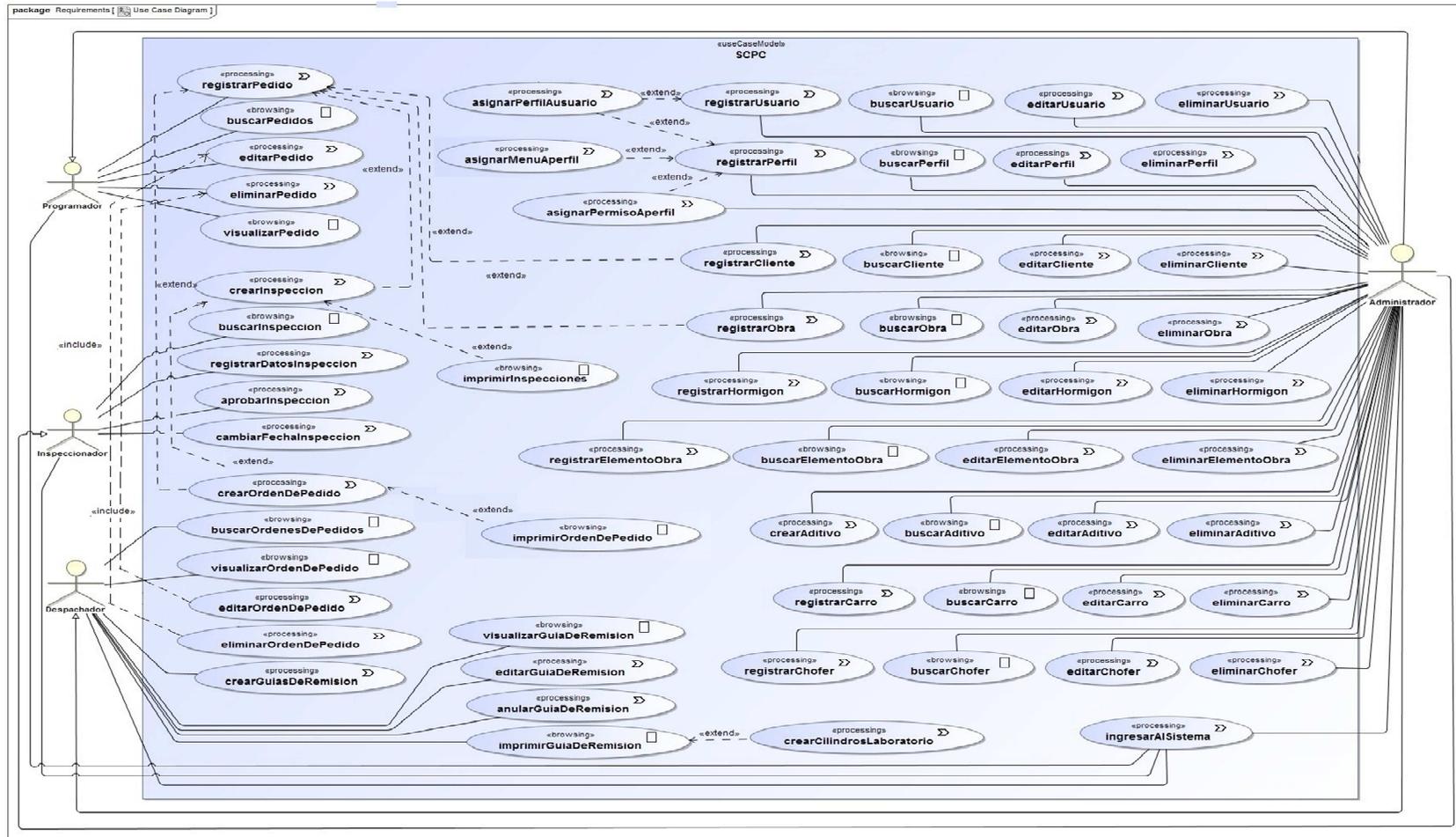


Figura 10 Casos de Uso del Sistema

4.1.3. DESCRIPCIÓN DE LOS CASOS DE USO

Tabla 56

Caso de Uso – ingresarAlSistema

Caso de Uso	ingresarAlSistema
Actores	Administrador, Programador, Inspeccionador y Despachador
Descripción	Los usuarios podrán ingresar al sistema con su respectivo nombre de usuario, contraseña y perfil.
Flujo básico	El usuario ingresa al sistema especificando su nombre de usuario, contraseña y perfil. El usuario hace clic en “Entrar”.
Precondiciones	El usuario debe estar registrado, tener asignado un perfil y permisos al perfil.
Post-condiciones	El usuario ingresa al sistema y lo puede utilizar.
Requerimientos Especiales	Ninguno.

Tabla 57

Caso de Uso – registrarUsuario

Caso de Uso	registrarUsuario
Actores	Administrador
Descripción	El administrador podrá registrar los datos de un usuario en el sistema.
Flujo básico	El administrador se dirige a la sección de usuarios y elige la opción “Crear”. Ingresa los datos en un formulario. Hace clic en “Guardar”.

Continúa →

Precondiciones	El administrador debe iniciar sesión en el sistema. La contraseña a registrarse debe ser igual a la de confirmación.
Post-condiciones	Los datos de un determinado usuario se registran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 58

Caso de Uso – buscarUsuario

Caso de Uso	buscarUsuario
Actores	Administrador
Descripción	El Administrador podrá buscar un determinado usuario, a través de varios atributos, ejemplo: cédula, apellido, etc.
Flujo básico	El administrador se dirige a la sección de usuarios y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: cédula, apellido, etc. Ingresa el valor del atributo y hace clic en “Buscar”.
Precondiciones	El administrador debe haber iniciado sesión en el sistema.
Post-condiciones	Los datos del usuario especificado se encuentran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 59

Caso de Uso – editarUsuario

Caso de Uso	editarUsuario
Actores	Administrador
Descripción	El administrador podrá editar los datos de un determinado usuario.
Flujo básico	<p>El administrador se dirige a la sección de usuarios y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: cédula, apellido, etc. Ingresar el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el usuario, el administrador hace clic en “Editar”.</p> <p>Los datos del usuario se muestran en un formulario. El administrador edita los datos.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el usuario para editar sus datos.
Post-condiciones	Los datos de un determinado usuario se editan con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 60

Caso de Uso – eliminarUsuario

Caso de Uso	eliminarUsuario
Actores	Administrador
Descripción	El administrador podrá eliminar todos los datos de un determinado usuario.
	Continúa →

Flujo básico	<p>El administrador se dirige a la sección de usuarios y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: cédula, apellido, etc.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el usuario, el administrador hace clic en “Eliminar”.</p> <p>El sistema muestra una ventana desplegable, para confirmar la acción.</p> <p>El administrador hace clic en “Eliminar”</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el usuario para eliminar sus datos.
Post-condiciones	El estado del usuario cambia para que sus datos no se muestren en el sistema.
Requerimientos Especiales	Los usuarios deben ser eliminados con autorización del gerente general de Hormigonera Equinoccial.

Tabla 61

Caso de Uso - registrarPerfil

Caso de Uso	registrarPerfil
Actores	Administrador
Descripción	El administrador podrá registrar los datos de un perfil en el sistema.
Flujo básico	<p>El administrador se dirige a la sección de perfiles y elige la opción “Registrar”.</p> <p>Ingresa los datos en un formulario.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema.
	Continúa →

Post-condiciones	Los datos de un determinado perfil se registran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 62

Caso de Uso – buscarPerfil

Caso de Uso	buscarPerfil
Actores	Administrador
Descripción	El Administrador podrá buscar un determinado perfil, a través de varios atributos, ejemplo: código, nombre.
Flujo básico	El administrador se dirige a la sección de perfiles y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre. Ingresa el valor del atributo y hace clic en “Buscar”.
Precondiciones	El administrador debe haber iniciado sesión en el sistema.
Post-condiciones	Los datos del perfil especificado se encuentran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 63

Caso de Uso – editarPerfil

Caso de Uso	editarPerfil
Actores	Administrador
Descripción	El administrador podrá editar los datos de un determinado perfil.
Continúa →	

Flujo básico	<p>El administrador se dirige a la sección de perfiles y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el perfil, el administrador hace clic en “Editar”.</p> <p>Los datos del perfil se muestran en un formulario.</p> <p>El administrador edita los datos.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el perfil para editar sus datos.
Post-condiciones	Los datos de un determinado perfil se editan con éxito.
Requerimientos Especiales	Ninguno.

Tabla 64

Caso de Uso – eliminarPerfil

Caso de Uso	eliminarPerfil
Actores	Administrador
Descripción	El administrador podrá eliminar todos los datos de un determinado perfil.
Flujo básico	<p>El administrador se dirige a la sección de perfiles y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el perfil, el administrador hace clic en “Eliminar”.</p> <p style="text-align: right;">Continúa →</p>

	El sistema muestra una ventana desplegable, para confirmar la acción. El administrador hace clic en “Eliminar”.
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el perfil para eliminar sus datos.
Post-condiciones	El estado del perfil cambia para que sus datos no se muestren en el sistema.
Requerimientos Especiales	Ninguno.

Tabla 65

Caso de Uso – asignarPerfilAusuario

Caso de Uso	asignarPerfilAusuario
Actores	Administrador
Descripción	El administrador podrá asignar un perfil a un determinado usuario.
Flujo básico	El administrador se dirige a la sección de perfiles y elige la opción “Usuario Perfil”. Hace clic en “Crear”. En una ventana desplegable escoge el perfil y escribe el apellido del usuario al que quiere asignar el perfil. Del cuadro de opciones que aparece, escoge el usuario deseado. Hace clic en “Guardar”.
Precondiciones	El administrador debe iniciar sesión en el sistema. El usuario seleccionado debe existir en el sistema.
Post-condiciones	El perfil se ha asignado a un usuario con éxito.
Requerimientos Especiales	Ninguno.

Tabla 66

Caso de Uso – asignarPermisoAperfil

Caso de Uso	asignarPermisoAperfil
Actores	Administrador
Descripción	El administrador podrá asignar uno o varios permisos a un determinado perfil.
Flujo básico	El administrador se dirige a la sección de perfiles y elige la opción "Acceso Permiso". Hace clic en "Crear". En una ventana desplegable escoge el permiso y el perfil al que quiere asignar el permiso. Hace clic en "Guardar".
Precondiciones	El administrador debe iniciar sesión en el sistema.
Post-condiciones	El permiso se han asignado a un perfil con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 67

Caso de Uso – asignarMenuAperfil

Caso de Uso	asignarMenuAperfil
Actores	Administrador
Descripción	El administrador podrá asignar uno o varios elementos del menú a un determinado perfil.
Flujo básico	El administrador se dirige a la sección de perfiles y elige la opción "Acceso Menú". Hace clic en "Crear". En una ventana desplegable escoge el menú y el perfil al que quiere asignar el menú. Hace clic en "Guardar".
	Continúa →

Precondiciones	El administrador debe iniciar sesión en el sistema. El administrador debe buscar el perfil y el elemento del menú.
Post-condiciones	El elemento del menú se han asignado a un perfil con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 68

Caso de Uso – registrarCliente

Caso de Uso	registrarCliente
Actores	Administrador, Programador
Descripción	El administrador y el Programador podrán registrar los datos de un cliente en el sistema.
Flujo básico	El administrador se dirige a la sección: Entidades/Clientes y elige la opción “Registrar Cliente”. Ingresa los datos en un formulario. Hace clic en “Guardar”.
Precondiciones	El administrador o Programador debe iniciar sesión en el sistema.
Post-condiciones	Los datos de un determinado cliente se registran con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 69

Caso de Uso – buscarCliente

Caso de Uso	buscarCliente
Actores	Administrador
Descripción	El Administrador podrá buscar un determinado cliente a través de varios atributos, ejemplo: cédula, apellido, etc.
Flujo básico	El administrador se dirige a la sección: Entidades/Clientes y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: cédula, apellido, etc. Ingresa el valor del atributo y hace clic en “Buscar”.
Precondiciones	El administrador debe haber iniciado sesión en el sistema.
Post-condiciones	Los datos del cliente especificado se encuentran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 70

Caso de Uso – editarCliente

Caso de Uso	editarCliente
Actores	Administrador
Descripción	El administrador podrá editar los datos de un determinado cliente.
Flujo básico	El administrador se dirige a la sección: Entidades/Clientes y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: cédula, apellido, etc. Ingresa el valor del atributo seleccionado y hace clic en “Aceptar “

Continúa →

	<p>Una vez encontrado el cliente, el administrador hace clic en “Editar”.</p> <p>Los datos del cliente se muestran en un formulario.</p> <p>El administrador edita los datos.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el cliente para editar sus datos.
Post-condiciones	Los datos de un determinado cliente se editan con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 71

Caso de Uso – eliminarCliente

Caso de Uso	eliminarCliente
Actores	Administrador
Descripción	El administrador podrá eliminar todos los datos de un determinado cliente.
Flujo básico	<p>El administrador se dirige a la sección: Entidades/Clientes y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: cédula, apellido, nombre.</p> <p>Ingresar el valor del atributo seleccionado y hace clic en “Aceptar”.</p> <p>Una vez encontrado el cliente, el administrador hace clic en “Eliminar”.</p> <p>El sistema muestra una ventana desplegable, para confirmar la acción.</p> <p>El administrador hace clic en “Eliminar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el cliente para eliminar sus datos. Continúa →

Post-condiciones	El estado del cliente cambia para que sus datos no se muestren en el sistema.
Requerimientos Especiales	Los clientes deben ser eliminados con autorización del gerente general de Hormigonera Equinoccial.

Tabla 72

Caso de Uso – registrarObra

Caso de Uso	registrarObra
Actores	Administrador, Programador
Descripción	El usuario podrá registrar los datos de una obra en el sistema.
Flujo básico	El usuario se dirige a la sección: Entidades/Obras y elige la opción “Crear”. Ingresa los datos en un formulario. Hace clic en “Guardar”.
Precondiciones	El administrador debe iniciar sesión en el sistema. Deben existir datos de clientes en el sistema.
Post-condiciones	Los datos de una determinada obra se registran con éxito.
Req. Especiales	Ninguno.

Tabla 73

Caso de Uso – buscarObra

Caso de Uso	buscarObra
Actores	Administrador
Descripción	El Administrador podrá buscar una determinada obra a través de varios atributos, ejemplo: código, nombre.
Continúa →	

Flujo básico	El usuario se dirige a la sección: Entidades/Obras y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre. Ingresa el valor del atributo y hace clic en “Buscar”.
Precondiciones	El administrador debe haber iniciado sesión en el sistema.
Post-condiciones	Los datos de una obra especificada se encuentran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 74

Caso de Uso – editarObra

Caso de Uso	editarObra
Actores	Administrador
Descripción	El administrador podrá editar los datos de una determinada obra.
Flujo básico	El usuario se dirige a la sección: Entidades/Obras y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre. Ingresa el valor del atributo seleccionado y hace clic en “Buscar”. Una vez encontrada la obra, el administrador hace clic en “Editar”. Los datos de la obra se muestran en un formulario. El administrador edita los datos. Hace clic en “Guardar”.
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar la obra para editar sus datos.
Continúa →	

Post-condiciones	Los datos de una determinada obra se editan con éxito.
Requerimientos Especiales	Ninguno.

Tabla 75

Caso de Uso – eliminarObra

Caso de Uso	eliminarObra
Actores	Administrador
Descripción	El administrador podrá eliminar todos los datos de una determinada obra.
Flujo básico	<p>El usuario se dirige a la sección: Entidades/Obras y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresar el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrada la obra, el administrador hace clic en “Eliminar”.</p> <p>El sistema muestra una ventana desplegable, para confirmar la acción.</p> <p>El administrador hace clic en “Eliminar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar la obra para eliminar sus datos.
Post-condiciones	El estado de la obra cambia para que sus datos no se muestren en el sistema.
Req. Especiales	Ninguno.

Tabla 76

Caso de Uso – registrarHormigon

Caso de Uso	registrarHormigon
Actores	Administrador
Descripción	El administrador podrá registrar los datos de un hormigón en el sistema.
Flujo básico	El usuario se dirige a la sección: Recursos/Hormigón y elige la opción “Crear”. Ingresa los datos en un formulario. Hace clic en “Guardar”.
Precondiciones	El administrador debe iniciar sesión en el sistema.
Post-condiciones	Los datos de un determinado hormigón se registran con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 77

Caso de Uso – buscarHormigon

Caso de Uso	buscarHormigon
Actores	Administrador
Descripción	El Administrador podrá buscar un determinado hormigón, a través de varios atributos, ejemplo: código, nombre.
Flujo básico	El usuario se dirige a la sección: Recursos/Hormigón y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre. Ingresa el valor del atributo y hace clic en “Buscar”.
Precondiciones	El administrador debe haber iniciado sesión en el sistema.
Continúa →	

Post-condiciones	Los datos del hormigón especificado se encuentran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 78

Caso de Uso – editarHormigon

Caso de Uso	editarHormigon
Actores	Administrador
Descripción	El administrador podrá editar los datos de un determinado hormigón.
Flujo básico	<p>El usuario se dirige a la sección: Recursos/Hormigón y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el hormigón, el administrador hace clic en “Editar”.</p> <p>Los datos del hormigón se muestran en un formulario.</p> <p>El administrador edita los datos.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el hormigón para editar sus datos.
Post-condiciones	Los datos de un determinado hormigón se editan con éxito.
Requerimientos Especiales	Los datos de un determinado hormigón deben ser editados con autorización del laboratorista de Hormigonera Equinoccial.

Tabla 79

Caso de Uso – eliminarHormigon

Caso de Uso	eliminarHormigon
Actores	Administrador
Descripción	El administrador podrá eliminar todos los datos de un determinado hormigón.
Flujo básico	<p>El usuario se dirige a la sección: Recursos/Hormigón y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el hormigón, el administrador hace clic en “Eliminar”.</p> <p>El sistema muestra una ventana desplegable, para confirmar la acción.</p> <p>El administrador hace clic en “Eliminar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el hormigón para eliminar sus datos.
Post-condiciones	El estado del hormigón cambia para que sus datos no se muestren en el sistema.
Requerimientos	Ninguno.
Especiales	

Tabla 80

Caso de Uso – registrarElementoObra

Caso de Uso	registrarElementoObra
Actores	Administrador
Descripción	El administrador podrá registrar los datos de un elemento en el sistema, ejemplo: pared, gradas, etc.
Flujo básico	El usuario se dirige a la sección: Recursos/Elemento Obra y elige la opción “Crear”. Ingresa los datos en un formulario. Hace clic en “Guardar”.
Precondiciones	El administrador debe iniciar sesión en el sistema.
Post-condiciones	Los datos de un determinado elemento se registran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 81

Caso de Uso – buscarElementoObra

Caso de Uso	buscarElementoObra
Actores	Administrador
Descripción	El Administrador podrá buscar un determinado elemento a través de varios atributos, ejemplo: código, nombre.
Flujo básico	El usuario se dirige a la sección: Recursos/Elemento Obra y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre. Ingresa el valor del atributo y hace clic en “Buscar”.
Precondiciones	El administrador debe haber iniciado sesión en el sistema.
Continúa →	

Post-condiciones	Los datos del elemento especificado se encuentran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 82

Caso de Uso – editarElementoObra

Caso de Uso	editarElementoObra
Actores	Administrador
Descripción	El administrador podrá editar los datos de un determinado elemento.
Flujo básico	<p>El usuario se dirige a la sección: Recursos/Elemento Obra y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el elemento, el administrador hace clic en “Editar”.</p> <p>Los datos del elemento se muestran en un formulario.</p> <p>El administrador edita los datos.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el elemento para editar sus datos.
Post-condiciones	Los datos de un determinado elemento se editan con éxito.
Requerimientos Especiales	Ninguno.

Tabla 83

Caso de Uso – eliminarElementoObra

Caso de Uso	eliminarElementoObra
Actores	Administrador
Descripción	El administrador podrá eliminar todos los datos de un determinado elemento.
Flujo básico	<p>El usuario se dirige a la sección: Recursos/Elemento Obra y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el elemento, el administrador hace clic en “Eliminar”.</p> <p>El sistema muestra una ventana desplegable, para confirmar la acción.</p> <p>El administrador hace clic en “Eliminar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el elemento para eliminar sus datos.
Post-condiciones	El estado del elemento cambia para que sus datos no se muestren en el sistema.
Requerimientos	Ninguno.
Especiales	

Tabla 84

Caso de Uso – crearAditivo

Caso de Uso	crearAditivo
Actores	Administrador
Descripción	El administrador podrá registrar los datos de un aditivo en el sistema.
Flujo básico	El usuario se dirige a la sección: Recursos/Aditivos y elige la opción “Crear”. Ingresa los datos en un formulario. Hace clic en “Guardar”.
Precondiciones	El administrador debe iniciar sesión en el sistema.
Post-condiciones	Los datos de un determinado aditivo se registran con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 85

Caso de Uso – buscarAditivo

Caso de Uso	buscarAditivo
Actores	Administrador
Descripción	El Administrador podrá buscar un determinado aditivo, a través de varios atributos, ejemplo: código, nombre.
Flujo básico	El usuario se dirige a la sección: Recursos/Aditivos y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre. Ingresa el valor del atributo y hace clic en “Buscar”.
Precondiciones	El administrador debe haber iniciado sesión en el sistema.
Continúa →	

Post-condiciones	Los datos del aditivo especificado se encuentran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 86

Caso de Uso – editarAditivo

Caso de Uso	editarAditivo
Actores	Administrador
Descripción	El administrador podrá editar los datos de un determinado aditivo.
Flujo básico	<p>El usuario se dirige a la sección: Recursos/Aditivos y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el aditivo, el administrador hace clic en “Editar”.</p> <p>Los datos del aditivo se muestran en un formulario.</p> <p>El administrador edita los datos.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el aditivo para editar sus datos.
Post-condiciones	Los datos de un determinado aditivo se editan con éxito.
Requerimientos Especiales	Ninguno.

Tabla 87

Caso de Uso – eliminarAditivo

Caso de Uso	eliminarAditivo
Actores	Administrador
Descripción	El administrador podrá eliminar todos los datos de un determinado aditivo.
Flujo básico	<p>El usuario se dirige a la sección: Recursos/Aditivos y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el aditivo, el administrador hace clic en “Eliminar”.</p> <p>El sistema muestra una ventana desplegable, para confirmar la acción.</p> <p>El administrador hace clic en “Eliminar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el aditivo para eliminar sus datos.
Post-condiciones	El estado del aditivo cambia para que sus datos no se muestren en el sistema.
Requerimientos	Ninguno.
Especiales	

Tabla 88

Caso de Uso – registrarCarro

Caso de Uso	registrarCarro
Actores	Administrador
Descripción	El administrador podrá registrar los datos de un carro en el sistema.
Flujo básico	El usuario se dirige a la sección: Recursos/Carros y elige la opción “Crear”. Ingresa los datos en un formulario. Hace clic en “Guardar”.
Precondiciones	El administrador debe iniciar sesión en el sistema.
Post-condiciones	Los datos de un determinado carro se registran con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 89

Caso de Uso – buscarCarro

Caso de Uso	buscarCarro
Actores	Administrador
Descripción	El Administrador podrá buscar un determinado carro, a través de varios atributos, ejemplo: código, nombre.
Flujo básico	El usuario se dirige a la sección: Recursos/Carros y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre. Ingresa el valor del atributo y hace clic en “Buscar”.
Precondiciones	El administrador debe haber iniciado sesión en el sistema.
Post-condiciones	Los datos del carro especificado se encuentran con éxito.
Req. Especiales	Ninguno.

Tabla 90

Caso de Uso – editarCarro

Caso de Uso	editarCarro
Actores	Administrador
Descripción	El administrador podrá editar los datos de un determinado usuario.
Flujo básico	<p>El usuario se dirige a la sección: Recursos/Carros y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el carro, el administrador hace clic en “Editar”.</p> <p>Los datos del carro se muestran en un formulario.</p> <p>El administrador edita los datos.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el carro para editar sus datos.
Post-condiciones	Administrador
Requerimientos	El administrador podrá editar los datos de un determinado
Especiales	carro.

Tabla 91

Caso de Uso – eliminarCarro

Caso de Uso	eliminarCarro
Actores	Administrador
Descripción	El administrador podrá eliminar todos los datos de un determinado carro.
Continúa →	

Flujo básico	<p>El usuario se dirige a la sección: Recursos/Carros y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: código, nombre.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el carro, el administrador hace clic en “Eliminar”.</p> <p>El sistema muestra una ventana desplegable para confirmar la acción.</p> <p>El administrador hace clic en “Eliminar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el carro para eliminar sus datos.
Post-condiciones	El estado del carro cambia para que sus datos no se muestren en el sistema.
Requerimientos Especiales	Ninguno.

Tabla 92

Caso de Uso – registrarChofer

Caso de Uso	registrarChofer
Actores	Administrador
Descripción	El administrador podrá registrar los datos de un chifer en el sistema.
Flujo básico	<p>El usuario se dirige a la sección: Recursos/Choferes y elige la opción “Crear”.</p> <p>Ingresa los datos en un formulario.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema.
	Continúa →

Post-condiciones	Los datos de un determinado chofer se registran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 93

Caso de Uso – buscarChofer

Caso de Uso	buscarChofer
Actores	Administrador
Descripción	El Administrador podrá buscar un determinado chofer, a través de varios atributos, ejemplo: cédula, apellido, etc.
Flujo básico	El usuario se dirige a la sección: Recursos/Choferes y elige la opción “Buscar”. En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: cédula, apellido, etc. Ingresa el valor del atributo y hace clic en “Buscar”.
Precondiciones	El administrador debe haber iniciado sesión en el sistema.
Post-condiciones	Los datos del chofer especificado se encuentran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 94

Caso de Uso – editarChofer

Caso de Uso	editarChofer
Actores	Administrador
Descripción	El administrador podrá editar los datos de un determinado chofer.
Continúa →	

Flujo básico	<p>El usuario se dirige a la sección: Recursos/Choferes y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: cédula, apellido, etc.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p> <p>Una vez encontrado el chofer, el administrador hace clic en “Editar”.</p> <p>Los datos del chofer se muestran en un formulario.</p> <p>El administrador edita los datos.</p> <p>Hace clic en “Guardar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el chofer para editar sus datos.
Post-condiciones	Los datos de un determinado chofer se editan con éxito.
Requerimientos Especiales	Ninguno.

Tabla 95

Caso de Uso – eliminarChofer

Caso de Uso	eliminarChofer
Actores	Administrador
Descripción	El administrador podrá eliminar todos los datos de un determinado chofer.
Flujo básico	<p>El usuario se dirige a la sección: Recursos/Choferes y elige la opción “Buscar”.</p> <p>En una ventana desplegable escoge un atributo para realizar la búsqueda, ejemplo: cédula, apellido, etc.</p> <p>Ingresa el valor del atributo seleccionado y hace clic en “Buscar”.</p>
	Continúa →

	<p>Una vez encontrado el chofer, el administrador hace clic en “Eliminar”.</p> <p>El sistema muestra una ventana desplegable, para confirmar la acción.</p> <p>El administrador hace clic en “Eliminar”.</p>
Precondiciones	El administrador debe iniciar sesión en el sistema y buscar el chofer para eliminar sus datos.
Post-condiciones	El estado del chofer cambia para que sus datos no se muestren en el sistema.
Requerimientos	Ninguno.
Especiales	

Tabla 96

Caso de Uso – registrarPedido

Caso de Uso	registrarPedido
Actores	Administrador, Programador
Descripción	El usuario podrá registrar los datos de un pedido en el sistema.
Flujo básico	<p>El usuario se dirige a la sección: Agenda de Pedidos y elige la opción “Registrar Pedido”.</p> <p>Ingresa los datos en un formulario.</p> <p>Hace clic en “Aceptar”.</p>
Precondiciones	El usuario debe iniciar sesión en el sistema.
Post-condiciones	Los datos de un determinado pedido se registran con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 97

Caso de Uso – buscarPedidos

Caso de Uso	buscarPedidos
Actores	Administrador, Programador
Descripción	El usuario podrá buscar varios pedidos programados en una fecha determinada.
Flujo básico	El usuario se dirige a la sección: Agenda de Pedidos y escoge una fecha. El sistema muestra todos los pedidos registrados en la fecha escogida.
Precondiciones	El usuario debe haber iniciado sesión en el sistema.
Post-condiciones	Según la fecha indicada, los pedidos se encuentran con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 98

Caso de Uso – editarPedido

Caso de Uso	editarPedido
Actores	Administrador, Programador
Descripción	El usuario podrá editar los datos de un determinado pedido.
Flujo básico	El usuario se dirige a la sección: Agenda de Pedidos y escoge una fecha. El sistema muestra todos los pedidos registrados en la fecha escogida. El usuario se ubica en el pedido que quiere editar y hace clic en “Editar Pedido”. Los datos del pedido se muestran en un formulario. El usuario edita los datos. Hace clic en “Aceptar”.
	Continua →

Precondiciones	El usuario debe iniciar sesión en el sistema y buscar el pedido a través de una fecha determinada.
Post-condiciones	Los datos de un determinado pedido se editan con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 99

Caso de Uso – eliminarPedido

Caso de Uso	eliminarPedido
Actores	Administrador, Programador
Descripción	El usuario podrá eliminar todos los datos de un determinado pedido.
Flujo básico	<p>El usuario se dirige a la sección: Agenda de Pedidos y escoge una fecha.</p> <p>El sistema muestra todos los pedidos registrados en la fecha escogida.</p> <p>El usuario se ubica en el pedido que quiere eliminar y hace clic en “Eliminar Pedido”.</p> <p>El sistema muestra una ventana desplegable, para confirmar la acción.</p> <p>El usuario hace clic en “Aceptar”.</p>
Precondiciones	El usuario debe iniciar sesión en el sistema y buscar el pedido a través de una fecha determinada.
Post-condiciones	El estado del pedido cambia para que sus datos no se muestren en el sistema.
Requerimientos	Ninguno.
Especiales	

Tabla 100

Caso de Uso – visualizarPedido

Caso de Uso	visualizarPedido
Actores	Administrador, Programador
Descripción	El usuario podrá visualizar todos los datos de un determinado pedido.
Flujo básico	<p>El usuario se dirige a la sección: Agenda de Pedidos y escoge una fecha.</p> <p>El sistema muestra todos los pedidos registrados en la fecha escogida.</p> <p>El usuario se ubica en el pedido que quiere visualizar y hace clic en “Visualizar Pedido”.</p> <p>El sistema muestra una ventana desplegable con todos los datos del pedido.</p>
Precondiciones	El usuario debe iniciar sesión en el sistema y buscar el pedido a través de una fecha determinada.
Post - condiciones	El usuario visualiza los datos del pedido respectivo.
Requerimientos Especiales	Ninguno.

Tabla 101

Caso de Uso – crearInspeccion

Caso de Uso	crearInspeccion
Actores	Administrador, Programador
Descripción	El usuario podrá crear una inspección para un determinado pedido.
Flujo básico	<p>Al momento de crear un pedido el usuario debe escoger la opción “Inspección = SI”.</p> <p style="text-align: right;">Continúa →</p>

	El sistema registrará de forma automática los datos básicos de una inspección, ejemplo: Código del pedido a inspeccionar, dirección, nombre del cliente, etc. Nota: Las inspecciones se crean un día antes de la fecha en la que se registró el pedido.
Precondiciones	El usuario debe iniciar sesión en el sistema. El usuario debe registrar un pedido con la opción “Inspección = SI”.
Post-condiciones	Los datos de una determinada inspección se registran con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 102

Caso de Uso – buscarInspeccion

Caso de Uso	buscarInspeccion
Actores	Administrador, Inspector
Descripción	El usuario podrá buscar una inspección a través de una fecha determinada.
Flujo básico	El usuario se dirige a la sección: Inspección e ingresa una fecha en un cuadro de texto. El sistema mostrará inspecciones de uno o varios pedidos.
Precondiciones	El usuario debe haber iniciado sesión en el sistema.
Post-condiciones	La inspección especificada se encuentra con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 103

Caso de Uso – registrarDatosInspeccion

Caso de Uso	registrarDatosInspeccion
Actores	Administrador, Inspector
Descripción	El usuario podrá registrar datos de una inspección realizada en una obra.
Flujo básico	El usuario se dirige a la sección: Inspección e ingresa una fecha en un cuadro de texto. El sistema mostrará inspecciones de uno o varios pedidos registrados un día después de la fecha ingresada. El usuario elige la inspección. Hace clic en “Editar Inspección”. El sistema muestra un formulario donde el usuario registra los datos capturados en la obra visitada. El usuario hace clic en “Aceptar”.
Precondiciones	El usuario debe iniciar sesión en el sistema.
Post-condiciones	Los datos de una determinada inspección se registran con éxito.
Requerimientos Especiales	Ninguno.

Tabla 104

Caso de Uso – aprobarInspeccion

Caso de Uso	aprobarInspeccion
Actores	Administrador, Inspector
Descripción	El usuario podrá aprobar la inspección realizada a una obra determinada.
Continúa →	

Flujo básico	<p>El usuario se dirige a la sección: Inspección e ingresa una fecha en un cuadro de texto. El sistema mostrará inspecciones de uno o varios pedidos registrados un día después de la fecha ingresada.</p> <p>El usuario elige la inspección. Hace clic en “Aprobar Inspección”.</p>
Precondiciones	El usuario debe iniciar sesión en el sistema.
Post-condiciones	Una inspección es aprobada con éxito.
Requerimientos Especiales	Ninguno.

Tabla 105

Caso de Uso – cambiarFechaInspeccion

Caso de Uso	cambiarFechaInspeccion
Actores	Administrador, Inspector
Descripción	El usuario podrá cambiar la fecha en la que se realizará una inspección.
Flujo básico	<p>El usuario se dirige a la sección: Inspección e ingresa una fecha en un cuadro de texto. El sistema mostrará inspecciones de uno o varios pedidos registrados un día después de la fecha ingresada”.</p> <p>El usuario elige la inspección. Hace clic en “Cambiar Fecha de Inspección”.</p> <p>El sistema muestra una ventana desplegable con una caja de texto.</p> <p>El usuario ingresa en la caja de texto la nueva fecha en la que se realizará el pedido.</p> <p>El usuario hace clic en “Aceptar”</p>

Continúa →

Precondiciones	El usuario debe haber iniciado sesión en el sistema.
Post-condiciones	La inspección especificada se ha modificado con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 106

Caso de Uso – imprimirInspecciones

Caso de Uso	imprimirInspecciones
Actores	Administrador, Inspector
Descripción	El usuario podrá imprimir varias inspecciones a través de una fecha determinada.
Flujo básico	El usuario se dirige a la sección: Inspección e ingresa una fecha en un cuadro de texto. El sistema mostrará inspecciones de uno o varios pedidos registrados un día después de la fecha ingresada”. El usuario hace clic en “Imprimir”.
Precondiciones	El usuario debe haber iniciado sesión en el sistema.
Post-condiciones	Las inspecciones se imprimen con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 107

Caso de Uso – crearOrdenDePedido

Caso de Uso	crearOrdenDePedido
Actores	Administrador, Programador
Descripción	El usuario podrá crear una Orden de Pedido en el sistema.
Flujo básico	<p>El usuario tiene dos opciones para crear una orden de pedido:</p> <ol style="list-style-type: none"> 1. Crear un Pedido con la opción: “Inspección = NO”. 2. Aprobar una Inspección. <p>Si el usuario realiza cualquiera de las dos opciones mencionadas, el sistema creará la Orden de Pedido de forma automática.</p>
Precondiciones	El usuario debe iniciar sesión en el sistema.
Post-condiciones	Los datos de una determinada Orden de Pedido se registran con éxito.
Requerimientos	El usuario debe crear un pedido escogiendo la opción
Especiales	“Inspección = NO” o aprobar una inspección.

Tabla 108

Caso de Uso – imprimirOrdenDePedido

Caso de Uso	imprimirOrdenDePedido
Actores	Administrador, Despachador
Descripción	El usuario podrá imprimir una Orden de Pedido.
Flujo básico	<p>El usuario se dirige a la sección: Despachos/”Doc. Orden de Pedido” e ingresa el código de la Orden de Pedido en un cuadro de texto.</p> <p>El sistema mostrará los datos de una Orden de Pedido en una plantilla.</p> <p>El usuario hace clic en “Imprimir”. Continúa →</p>

Precondiciones	El usuario debe haber iniciado sesión en el sistema.
Post-condiciones	Los datos de la Orden de Pedido especificada se imprime con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 109

Caso de Uso – buscarOrdenesDePedido

Caso de Uso	buscarOrdenesDePedidos
Actores	Administrador, Despachador
Descripción	El usuario podrá buscar varias Órdenes de Pedidos programadas en una fecha determinada.
Flujo básico	El usuario se dirige a la sección: Despachos/"Agenda de O. Pedido" y escoge una fecha. El sistema muestra todas las Órdenes de Pedidos registradas en la fecha escogida.
Precondiciones	El usuario debe haber iniciado sesión en el sistema.
Post-condiciones	Según la fecha indicada, las Órdenes de Pedidos se encuentran con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 110

Caso de Uso – visualizarOrdenDePedido

Caso de Uso	visualizarOrdenDePedido
Actores	Administrador, Despachador
Descripción	El usuario podrá visualizar todos los datos de una determinada Orden de Pedido.
Flujo básico	El usuario se dirige a la sección: Despachos/"Agenda de O. Pedido" y escoge una fecha. El sistema muestra todas las Órdenes de Pedidos registradas en la fecha escogida. El usuario se ubica en la Orden de Pedido que quiere visualizar y hace clic en "Visualizar Orden de Pedido". El sistema muestra una ventana desplegable con todos los datos de la Orden de Pedido.
Precondiciones	El usuario debe iniciar sesión en el sistema y buscar la Orden de Pedido a través de una fecha determinada.
Post-condiciones	El usuario visualiza los datos de la orden respectiva.
Requerimientos Especiales	Ninguno.

Tabla 111

Caso de Uso – editarOrdenDePedido

Caso de Uso	editarOrdenDePedido
Actores	Administrador, Despachador
Descripción	El usuario podrá editar los datos de una determinada Orden de Pedido.
Flujo básico	Una determinada Orden de Pedido se modificará si y sólo si se modifica un Pedido.

Continúa →

Precondiciones	El usuario debe iniciar sesión en el sistema.
Post-condiciones	Los datos de una determinada orden de pedido se editan con éxito.
Requerimientos Especiales	Ninguno.

Tabla 112

Caso de Uso – eliminarOrdenDePedido

Caso de Uso	eliminarOrdenDePedido
Actores	Administrador, Despachador
Descripción	El usuario podrá eliminar todos los datos de una Orden de Pedido.
Flujo básico	Una determinada Orden de Pedido se eliminará si y sólo si se elimina un Pedido.
Precondiciones	El usuario debe iniciar sesión en el sistema.
Post-condiciones	El estado de la Orden de Pedido cambia para que sus datos no se muestren en el sistema.
Requerimientos Especiales	Ninguno.

Tabla 113

Caso de Uso – crearGuiasDeRemision

Caso de Uso	crearGuiasDeRemision
Actores	Administrador, Despachador
Descripción	El usuario podrá crear guías de remisión con datos básicos como “dirección de la obra, volumen y número de viaje”.
Flujo básico	<p>El usuario se dirige a la sección: Despachos/”Agenda de O. Pedido” y escoge una fecha.</p> <p>El sistema muestra todas las Órdenes de Pedidos registradas en la fecha escogida.</p> <p>El usuario se ubica en una orden de pedido y hace clic en “Crear Guías de Remisión”.</p> <p>El sistema muestra una ventana desplegable con un número de guías equivalente al total del volumen solicitado dividido para la capacidad de transporte de los carros (8 m3).</p>
Precondiciones	El usuario debe iniciar sesión en el sistema y buscar la Orden de Pedido a través de una fecha determinada.
Post-condiciones	Las guías de remisión se crean con éxito.
Requerimientos	Ninguno.
Especiales	

Tabla 114

Caso de Uso – visualizarGuiaDeRemision

Caso de Uso	visualizarGuiaDeRemision
Actores	Administrador, Despachador
Descripción	El usuario podrá visualizar todos los datos de una guía de remisión.
Flujo básico	<p>El usuario se dirige a la sección: Despachos/”Agenda de O. Pedido” y escoge una fecha.</p> <p>El sistema muestra todas las Órdenes de Pedidos registradas en la fecha escogida. Continúa →</p>

	<p>El usuario se ubica en la Orden de Pedido y hace clic en “Ver/Crear Guías de Remisión”.</p> <p>El sistema muestra una ventana desplegable con las guías de remisión pertenecientes al pedido seleccionado.</p> <p>El usuario selecciona una guía de remisión y hace clic en “Visualizar Guía de Remisión”.</p>
Precondiciones	El usuario debe iniciar sesión en el sistema y buscar la Orden de Pedido que contiene la Guía de Remisión que desea visualizar.
Post-condiciones	El usuario visualiza los datos de la Guía de Remisión.
Requerimientos	Ninguno.
Especiales	

Tabla 115

Caso de Uso – editarGuiaDeRemision

Caso de Uso	editarGuiaDeRemision
Actores	Administrador, Despachador
Descripción	El usuario podrá editar los datos de una guía de remisión.
Flujo básico	<p>El usuario se dirige a la sección: Despachos/”Agenda de O. Pedido” y escoge una fecha.</p> <p>El sistema muestra todas las Órdenes de Pedidos registradas en la fecha escogida.</p> <p>El usuario se ubica en la Orden de Pedido y hace clic en “Ver/Crear Guías de Remisión”.</p> <p>El sistema muestra una ventana desplegable con las guías de remisión pertenecientes al pedido seleccionado.</p> <p>El usuario selecciona una guía de remisión y hace clic en “Editar Guía de Remisión”.</p>

Continúa →

Precondiciones	El usuario debe iniciar sesión en el sistema y buscar la Orden de Pedido que contiene la Guía de Remisión que desea editar.
Post-condiciones	El usuario edita los datos de la Guía de Remisión con éxito.
Requerimientos Especiales	Para la asignación de recursos a una guía, primero verificar la disponibilidad de los mismos.

Tabla 116

Caso de Uso – anularGuiaDeRemision

Caso de Uso	anularGuiaDeRemision
Actores	Administrador, Despachador
Descripción	El usuario podrá eliminar todos los datos de una Orden de Pedido.
Flujo básico	<p>El usuario se dirige a la sección: Despachos/"Agenda de O. Pedido" y escoge una fecha.</p> <p>El sistema muestra todas las Órdenes de Pedidos registradas en la fecha escogida.</p> <p>El usuario se ubica en la orden de pedido y hace clic en "Ver/Crear Guías de Remisión".</p> <p>El sistema muestra una ventana desplegable con las guías de remisión pertenecientes al pedido seleccionado.</p> <p>El usuario selecciona una guía de remisión y hace clic en "Anular Guía de Remisión".</p>
Precondiciones	El usuario debe iniciar sesión en el sistema y buscar la Orden de Pedido que contiene la Guía de Remisión que desea anular.
Post-condiciones	El estado de la Guía de Remisión cambia para que sus datos no se muestren en el sistema.
Requerimientos Especiales	Ninguno.

Tabla 117

Caso de Uso – imprimirGuiaDeRemision

Caso de Uso	imprimirGuiaDeRemision
Actores	Administrador, Despachador
Descripción	El usuario podrá imprimir los datos de una Guía de Remisión.
Flujo básico	<p>El usuario se dirige a la sección: Despachos/"Agenda de O. Pedido" y escoge una fecha.</p> <p>El sistema muestra todas las Órdenes de Pedidos registradas en la fecha escogida.</p> <p>El usuario se ubica en la orden de pedido y hace clic en "Ver/Crear Guías de Remisión".</p> <p>El sistema muestra una ventana desplegable con las guías de remisión pertenecientes al pedido seleccionado.</p> <p>El usuario selecciona una guía de remisión y hace clic en "Imprimir Guía de Remisión".</p>
Precondiciones	El usuario debe iniciar sesión en el sistema y buscar la Orden de Pedido que contiene la Guía de Remisión que desea imprimir.
Post-condiciones	La Guía de Remisión se imprime con éxito.
Requerimientos Especiales	Verificar que la hoja donde se va a imprimir contenga el mismo número de guía registrado en el sistema.

Tabla 118

Caso de Uso – crearCilindrosLaboratorio

Caso de Uso	crearCilindrosLaboratorio
Actores	Administrador, Despachador
Descripción	El usuario podrá registrar datos del hormigón tomado de un despacho en forma de cilindro.
Flujo básico	<p>El usuario se dirige a la sección: Despachos/"Agenda de O. Pedido" y escoge una fecha. El sistema muestra todas las Órdenes de Pedidos registradas en la fecha escogida.</p> <p>El usuario se ubica en la orden de pedido y hace clic en "Crear Cilindros".</p> <p>El sistema muestra una ventana desplegable para ingresar el número de cilindros (máximo 4: 2 para el laboratorio local y/o 2 para enviar al cliente) que se desea tomar de un despacho.</p> <p>El usuario selecciona el número y hace clic en "Aceptar".</p>
Precondiciones	El usuario debe iniciar sesión en el sistema y buscar la Orden de Pedido que contiene la Guía de Remisión de la cual se desea tomar las muestras de hormigón (crear cilindros).
Post-condiciones	Los cilindros son creados con éxito.
Requerimientos	Preguntar cuántos cilindros desea que se le envíe al cliente
Especiales	como muestras del hormigón preparado.

4.2. DISEÑO CONCEPTUAL.

La Figura 11 muestra el diagrama de clases de la aplicación, la misma que está basado en el análisis de requisitos del paso anterior.

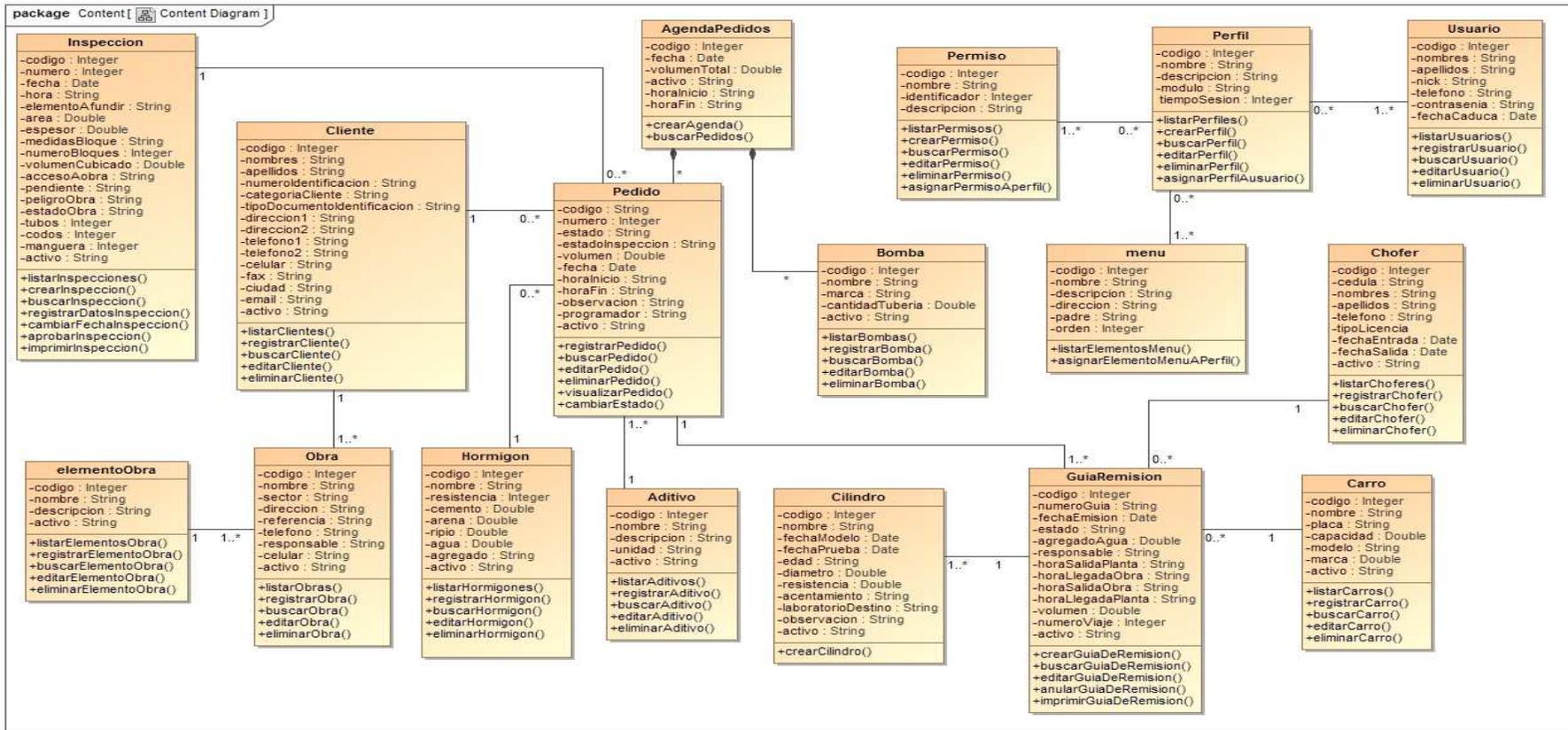


Figura 11 Diagrama de Clases del Sistema

4.3. DISEÑO DE NAVEGACIÓN.

En la Figura 12 se puede observar el diseño de la estructura de navegación en el sistema, el mismo que se implementará en la aplicación

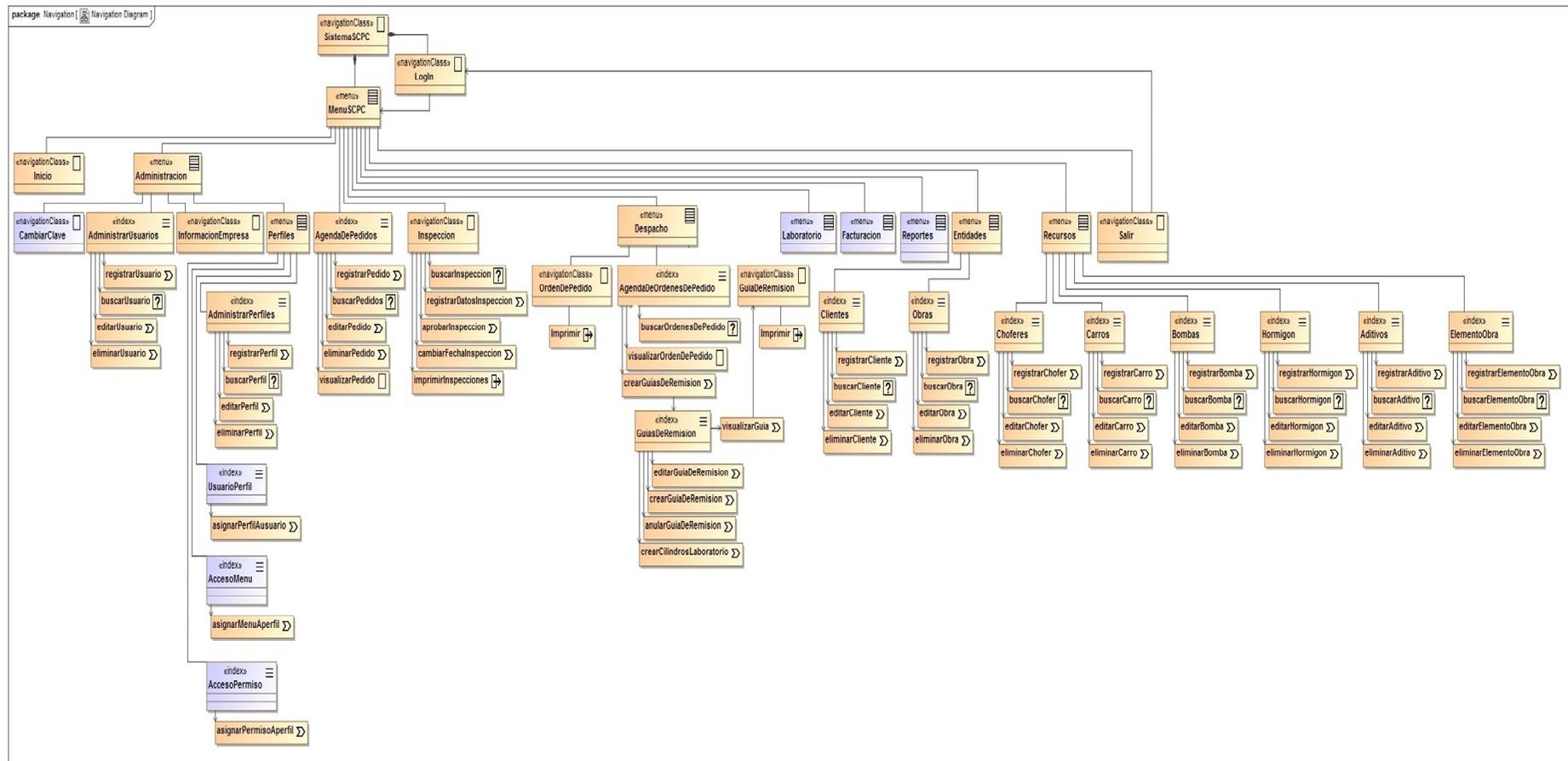


Figura 12 Diseño de Navegación del Sistema

Nota: Los elementos con fondo de color azul no están considerados dentro del alcance del presente proyecto.

4.4. DISEÑO DE PRESENTACIÓN.

El diseño de presentación está basado en la información obtenida en el análisis de requisitos y en el diseño de navegación. Este diseño está dividido en 6 partes:

- Página genérica
- Información de la Empresa
- Agenda de Pedidos
- Inspección
- Agenda de Despachos
- Documentos: “Orden de Pedido” y “Guía de Remisión”

4.4.1. PÁGINA GENÉRICA

La Figura 13 muestra un diseño genérico que se usará en las siguientes páginas:

- Administrar Usuarios
- Administrar Perfiles
- Usuario Perfil
- Acceso Menú
- Acceso Permiso
- Clientes
- Obras
- Choferes
- Carros
- Hormigón
- Aditivos
- Elementos Obra

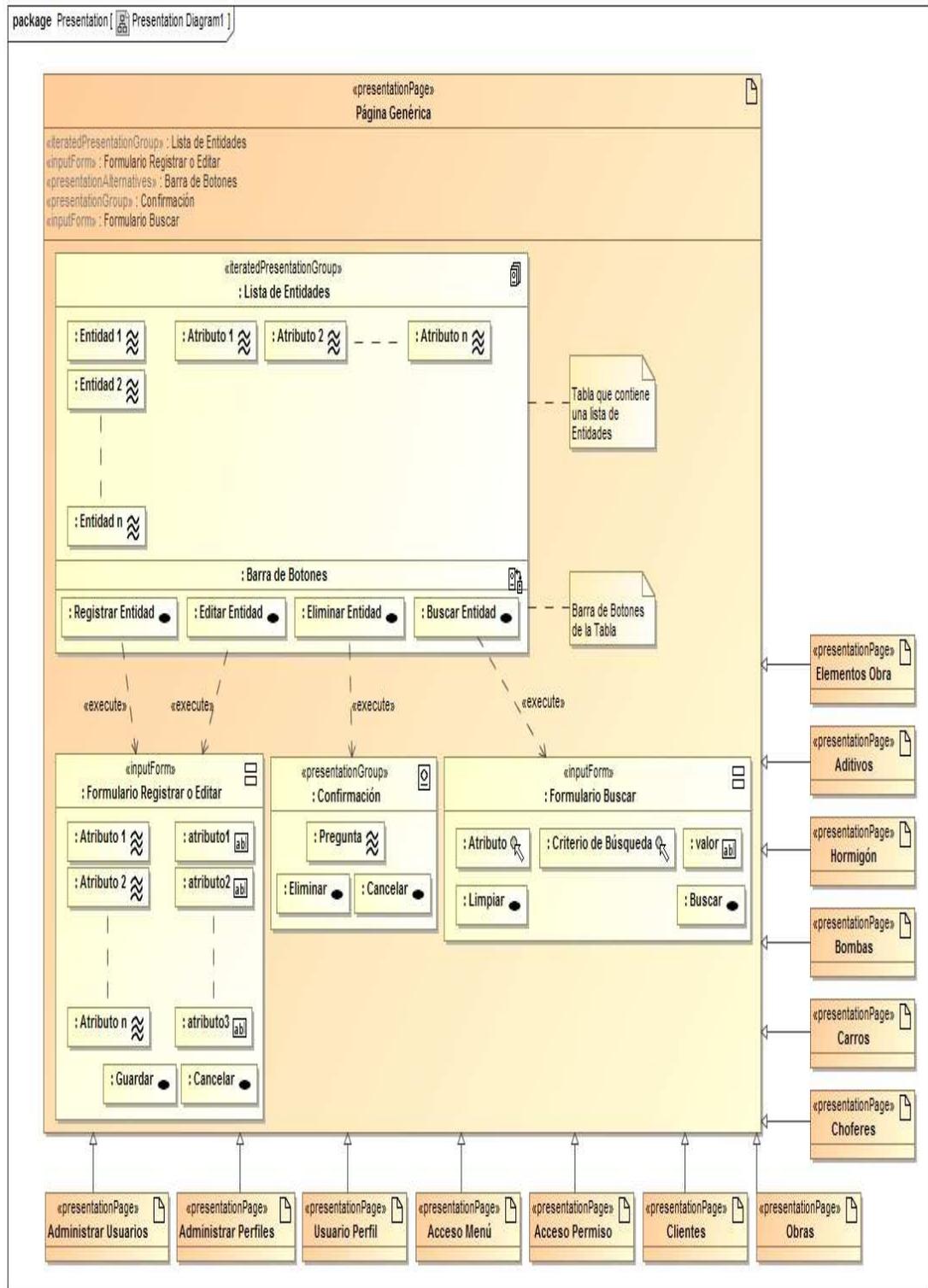


Figura 13 Página Genérica

4.4.2. INFORMACIÓN DE LA EMPRESA

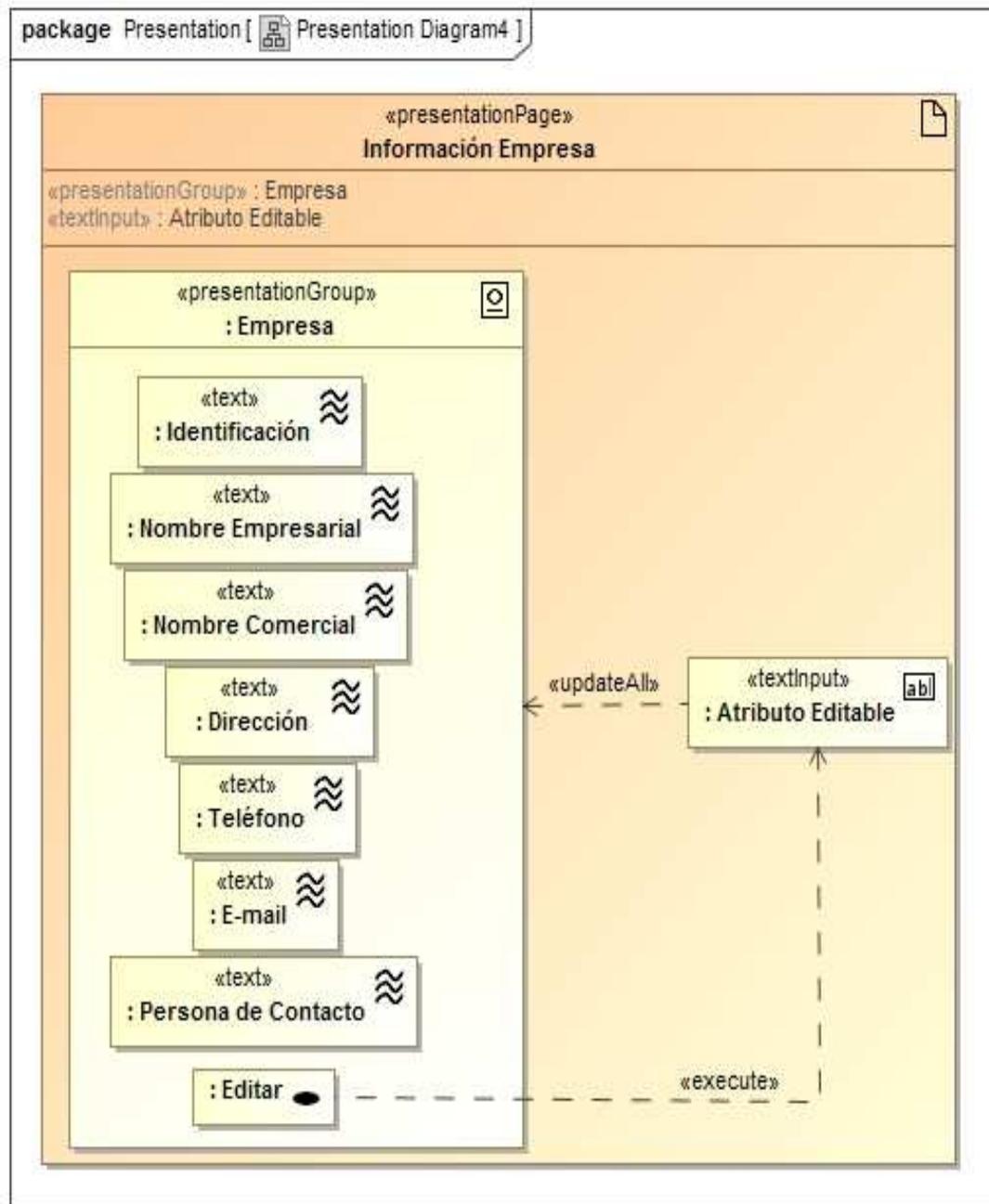


Figura 14 Página: Información Empresa

4.4.3. AGENDA DE PEDIDOS

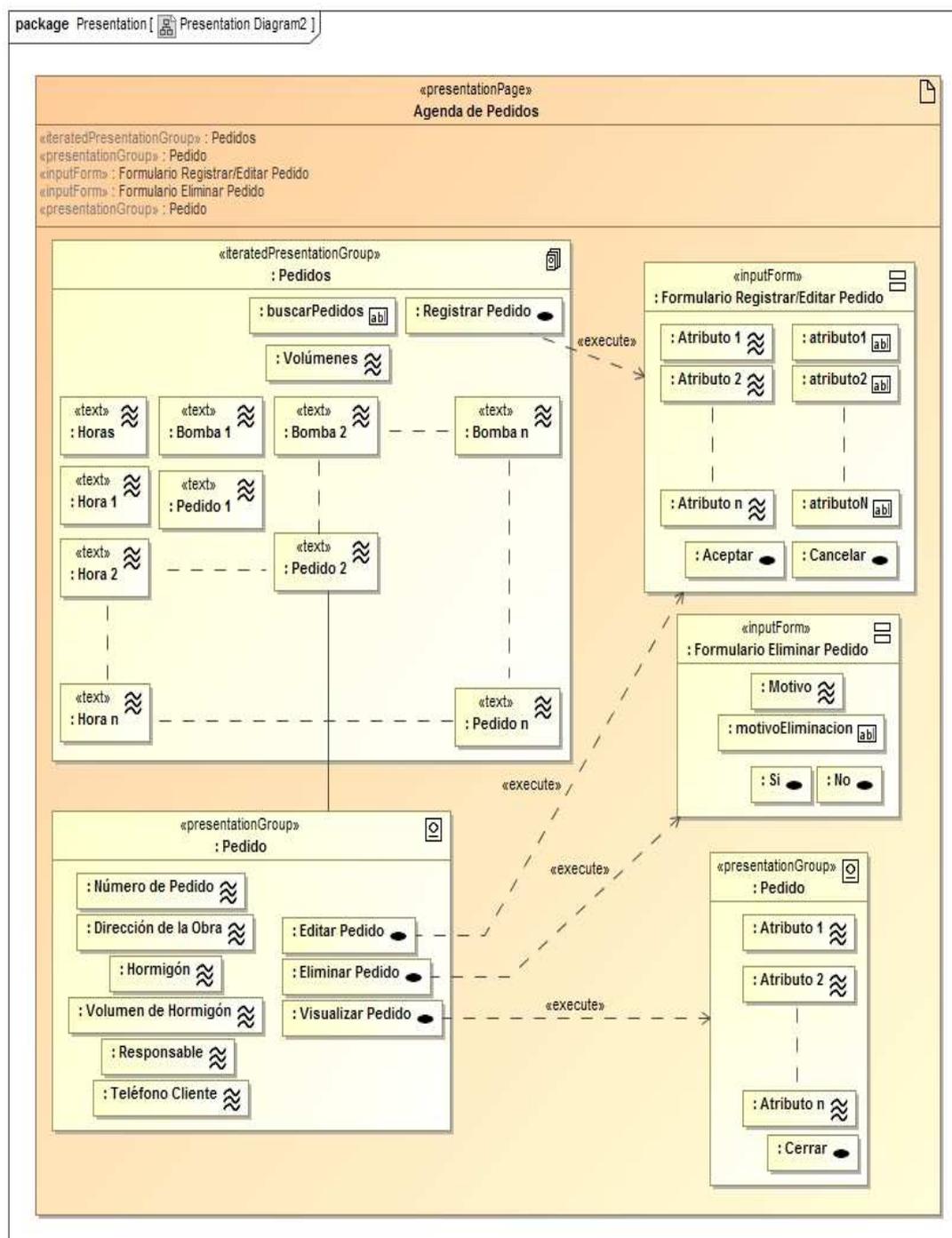


Figura 15 Página: Agenda de Pedidos

4.4.4. INSPECCIÓN

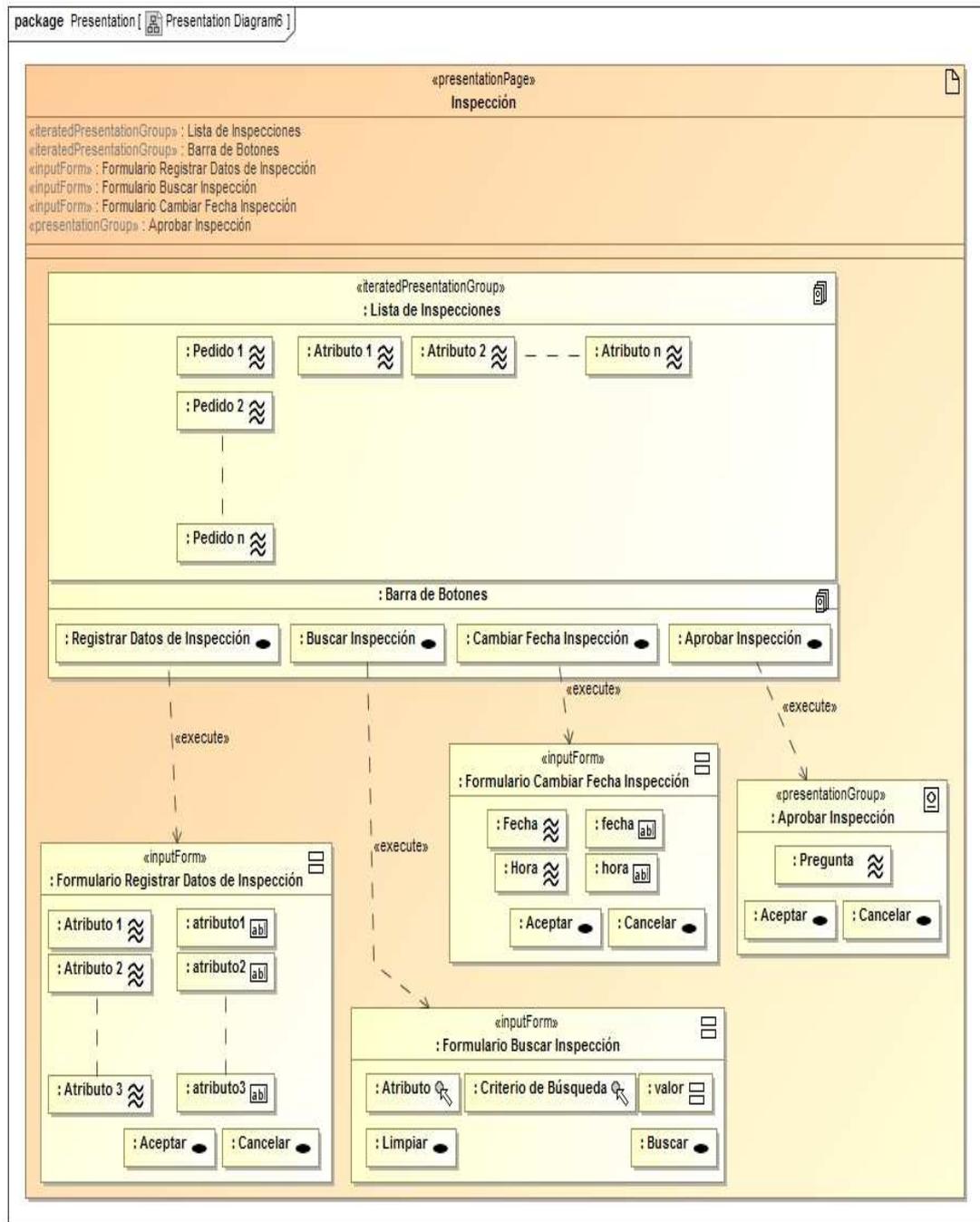


Figura 16 Página: Inspección

4.4.5. AGENDA DE DESPACHOS

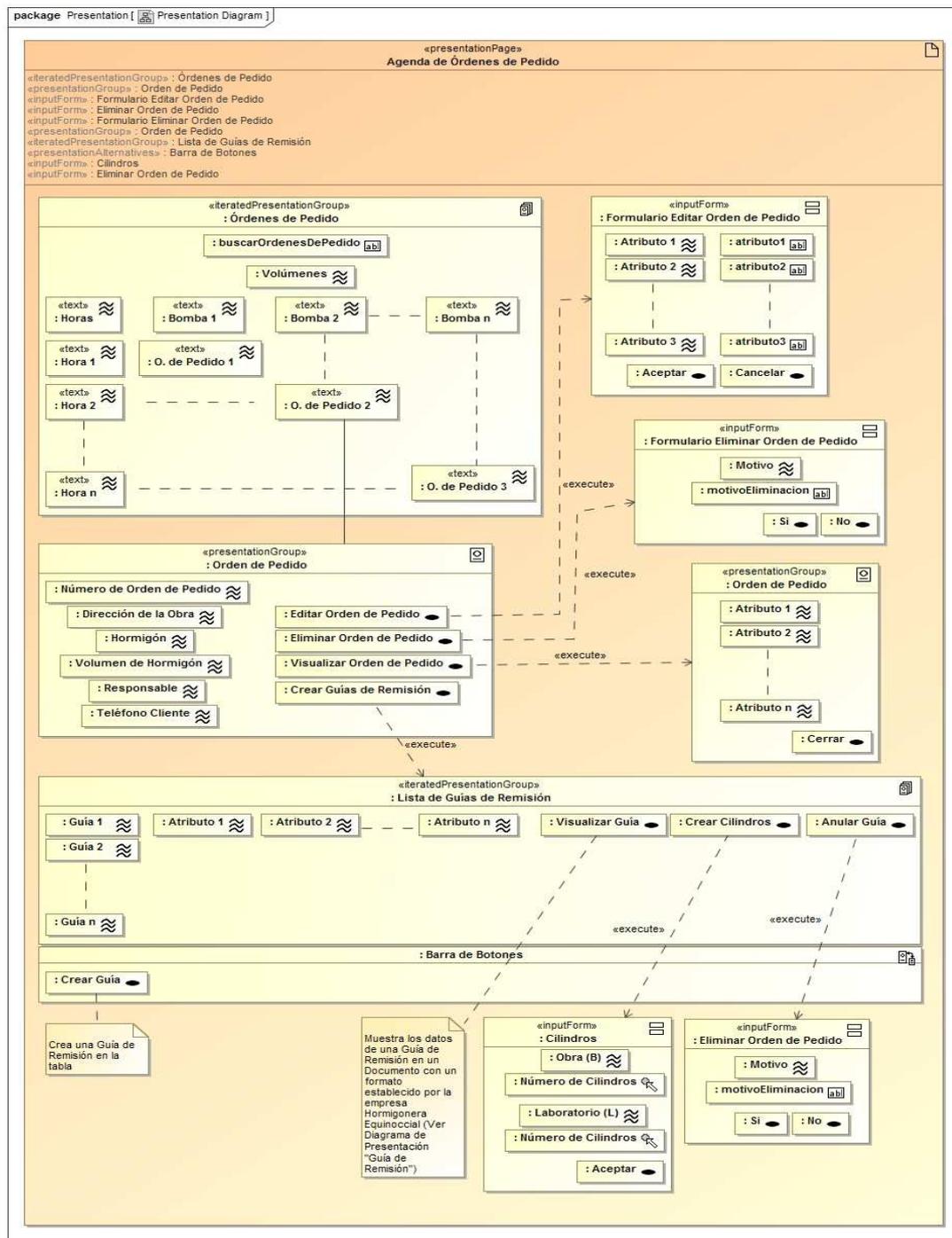


Figura 17 Página: Agenda de Órdenes de Pedido

4.4.6. DOCUMENTOS: “ORDEN DE PEDIDO” Y “GUÍA DE REMISIÓN”

package Presentation [Presentation Diagram7]

«presentationPage»
Doc. Orden de Pedido

HORMIGONERA EQUINOCCIAL
CONTRIBUYENTE ESPECIAL SEGUN LA RESOLUCION No. 194
RUC: 1791287851001

Calle Chedak 100 y Av. Eloy Alfaro Norte
Telf: 2483-652 / 2483-653 / 2483-616 Fax: 2483-607

ORDEN DE PEDIDO - V003 N° 000188

Fecha: _____

CLIENTE	Hormigón
Nombres: _____ Apellidos: _____ C.I./R.U.C.: _____ Dirc. Dom./Ofic.: _____ (Calle Banco Sector): _____ Telf. Dom.: _____ Telf. Ofic.: _____ Telf. Móvil: _____ Fax: _____	Volumen (m ³): _____ f.c.: _____ P.U.: _____ Activo: <input type="checkbox"/> Sí <input type="checkbox"/> No Bomba: <input type="checkbox"/> Sí <input type="checkbox"/> No Total O.P. \$: _____ Condiciones de pago: _____

OBRA	Abonos																
Nombre Obra: _____ Nombre Residente: _____ Dirc. Obra: _____ Telf. de la Obra: _____ Telf. Móvil: _____ Fecha aproximada de Fundición: _____ Hora aproximada de Fundición: _____ Elementos a Fundir: _____ Croquis: _____	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Abono N°</th> <th># Recibo</th> <th>Fecha</th> <th>Cobrador</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table> Observaciones: _____	Abono N°	# Recibo	Fecha	Cobrador												
Abono N°	# Recibo	Fecha	Cobrador														

Hormigonera Equinoccial Autorizado por _____ Cliente: _____

«presentationPage»
Doc. Guía de Remisión

HORMIGONERA EQUINOCCIAL
CONTRIBUYENTE ESPECIAL SEGUN LA RESOLUCION No. 194
RUC: 1791287851001
001-001-00 N° 0078512
AUTORIZACIÓN SRI: 1114059006

Calle Chedak 100 y Av. Eloy Alfaro - La Chorrera
Telf: (02) 2483-652 / 2483-653 / 2483-616 / 2483-607 / 2809-639
Cable: Equinocc

Datos Transporte	Datos Hormigón	
Fecha: _____	Viaje #: _____	Volumen (m ³): _____
Cliente: _____	Carru #: _____	% a 68 28 días = _____ Kg/cm ²
Obra: _____	Chofer: _____	Aditivo: _____
Orcección: _____	Sello #: _____	Agregado (T. Max): _____
	Bomba: _____	

Elemento Fundido: _____

Hora de salida de la Planta: _____ Hora de llegada a la Obra: _____
 Hora de llegada a la Planta: _____ Hora de salida de la Obra: _____

CONTROL DESPACHO CLIENTE		
PEDIDO	DESPACHADO	SALDO

TIEMPO MAXIMO DE DESCARGA DEL MIXER: 45 MINUTOS

Observaciones: _____

NOTA: Hormigonera Equinoccial no es responsable por la calidad del hormigón si el cliente le aumenta agua, aditivos, etc. o si se excede de 45 minutos en su descarga.

Agua en Obra: _____ Lts.
Autorizado Por: _____

Recibido Por:	Cobrador de H.E. en Obra:	Despachado Por:
Firma: _____	Firma: _____	Firma: _____
RUC/C.I.: _____	Nombre: _____	Nombre: _____

HORMIGONERA EQUINOCCIAL S.A. - EMPRESA DEL COMERCIO - SECTOR INDUSTRIAL - CALLE CHEDAK 100 Y AV. ELOY ALFARO NORTE - LA CHORRERA - PANAMA
 FORM DE CARTELO - HORMIGONERAS

Figura 18 Páginas: Orden de Pedido y Guía de Remisión

Fuente: Hormigonera Hequinoccial

4.5. DISEÑO DE PROCESOS.

La Figura 19 muestra un diseño genérico sobre las operaciones básicas realizadas en el desarrollo del sistema "SCPC".

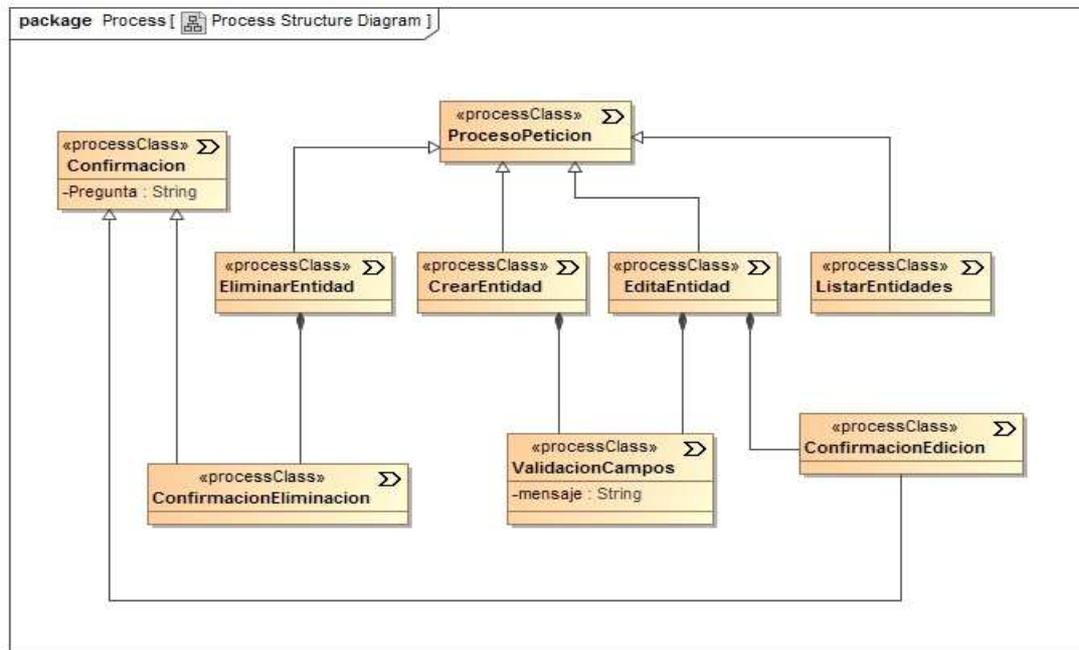


Figura 19 Modelo de la estructura de procesos

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES.

- La demanda de aplicaciones web ha hecho que las herramientas para la creación de software evolucionen sin limitaciones, cada instrumento ofrece un gran número de funcionalidades y facilidades para su uso, pero es importante recordar que una correcta evaluación garantiza una selección óptima para satisfacer las necesidades del usuario.
- El estándar ISO 25000 ha sido un elemento fundamental para la construcción del modelo de evaluación de frameworks, ya que los factores de calidad que propone la norma se ajustan al desarrollo de los modelos mixtos, guiando al usuario en la elección de atributos de calidad de los frameworks para su posterior evaluación.
- Tomando como referencia el modelo de calidad presentado por la Norma ISO 25000 y siguiendo las directrices y técnicas que proporciona el método IQMC, se identificaron características, subcaracterísticas, atributos de calidad y métricas necesarias para el desarrollo del modelo de calidad resultante.
- Después de aplicar el modelo de evaluación, se observó que jQuery supera a MooTools en la mayoría de las características, destacándose la Fiabilidad de la herramienta debido a la gran comunidad que respalda su evolución.
- La extensa cantidad de: documentación, foros, tutoriales y componentes prefabricados, han hecho que jQuery sea fácil de aprender y usar. Esta herramienta se integra bastante bien con una

gran cantidad de tecnologías orientadas al desarrollo de aplicaciones web, lo cual ha beneficiado a desarrolladores independientes y empresas al momento de crear aplicaciones de vanguardia.

- Para realizar el modelado del caso práctico se utilizó la metodología UWE (UML Web Engineering), debido a que abarca áreas relacionadas con la Web como la navegación, presentación, los procesos de negocio y los aspectos de adaptación. Esta metodología hace énfasis en el desarrollo de los diagramas de navegación y presentación, ya que estos permiten estructurar, entender y formalizar el dominio del problema como las decisiones de diseño de un sitio web.
- Después de desarrollar la aplicación SCPC (Sistema de Comercialización y Producción de Concreto) se observó que jQuery es una herramienta concreta porque provee un conjunto de componentes específicos necesarios, y completa porque dependiendo de la escala, dificultad y enfoque del proyecto propuesto se acopla a las necesidades generadas en el transcurso del desarrollo.

5.2. RECOMENDACIONES.

- Para construir un modelo de evaluación de software robusto, flexible y escalable, se recomienda enfocarse en los modelos mixtos y hacer uso de los factores de calidad propuestos por la norma ISO 25000, ya que estos en conjunto presentan una amplia lista de factores de calidad esenciales en un producto software.
- Si se desea crear un modelo de evaluación utilizando el método IQMC, se recomienda no obviar ningún paso, ya que en cada uno se reúne información precisa para crear un modelo de evaluación robusto y concreto, evitando factores de calidad abstractos.

Se recomienda utilizar el modelo de evaluación desarrollado en este proyecto cuando el usuario necesite seleccionar la mejor opción entre varios frameworks orientados al Front-End de una aplicación web, este modelo posee características concretas y fáciles de medir.

- Para facilitar el desarrollo de una aplicación web, se recomienda usar la metodología UWE, ya que esta se basa en el proceso unificado y UML facilitando la visualización, especificación, construcción y documentación de software. UWE recomienda la utilización de MagicDraw para la construcción de diagramas.
- Sin importar el nivel de experiencia en el uso de JavaScript, se recomienda utilizar jQuery para el desarrollo de aplicaciones orientados a la web, ya que este posee una API extensa y fácil de aprender, apoyada con una gran cantidad de tutoriales, libros y foros.

Bibliografía

Álvarez, M. A. (11 de 01 de 2010). *DesarrolloWeb*. Recuperado el 24 de 02 de 2014, de DesarrolloWeb: <http://www.desarrolloweb.com/>

APPERS, W. (2008). *WEB APPERS*. Recuperado el 13 de 10 de 2013, de WEB APPERS: <http://matthiasschuetz.com/javascript-framework-matrix/en/>

Campderrich, B. (2003). *Ingeniería de Software*. Aragón: UOC.

Castro, J. (01 de 01 de 2014). Cómo Evaluar un lenguaje de Programación. (M. d. Web, Entrevistador)

Coral Calero, Á. M. (2010). *Calidad del producto y proceso software*. Madrid: Ra-Ma.

Dave Methvin, R. W. (26 de 08 de 2006). *jQuery*. Recuperado el 02 de 09 de 2013, de jQuery: <http://jquery.com/>

Debrauwer, L., & Van delLeyde, F. (2005). *UML 2: iniciación, ejemplos y ejercicios corregidos*. Barcelona: ENI.

ISO. (01 de 01 de 2005). *ISO25000*. Recuperado el 25 de 10 de 2013, de ISO25000: <http://iso25000.com/>

jQuery. (2014). *jQuery*. Obtenido de jQuery: <http://jquery.com/>

Ludwig. (01 de 03 de 2009). *uwe*. Recuperado el 13 de 10 de 2013, de uwe: <http://uwe.pst.ifi.lmu.de/infoMAEWA.html>

MooTools. (2014). *MooTools Web*. Obtenido de MooTools Web: <http://mootools.net/>

Mozilla, D. (01 de 01 de 2005). *MDN*. Recuperado el 12 de 02 de 2014, de MDN:https://developer.mozilla.org/es/docs/Introducci%C3%B3n_a_JavaScript_orientado_a_objetos

MySQL. (s.f.). *About MySQL*. Recuperado el 10 de Noviembre de 2013, de MySQL: <http://www.mysql.com/>

Nyman, R. (13 de 10 de 2013). *robertnyman*. Recuperado el 13 de 10 de 2013, de robertnyman: <http://robertnyman.com/javascript/#javascript-1-7-compatibility>

Osmani, A. (2012). *JavaScript Design Patterns*. Londres: O'Reilly Media.

Pressman, R. (2010). *Ingeniería de Software Un enfoque práctico*. McGraw Hill.

Radcliffe, M. (2014). *Open Source Initiative*. Recuperado el 21 de 02 de 2014, de Open Source Initiative: <http://opensource.org/licenses/MIT>

Rivera, F. L. (2008). *Lógica y programación orientada a objetos*. Bogotá: ITM.

Tavárez, D. (2009). *Maestros del Web*. Recuperado el 01 de 10 de 2013, de Maestros del Web:

<http://www.maestrosdelweb.com/editorial/comparacion-frameworks-javascript/>

Valerio Proietti, D. W. (01 de 01 de 2006). *MooTools*. Recuperado el 12 de 09 de 2013, de MooTools: <http://mootools.net/>

Vargas, Z. R. (08/06/2009). *LA INVESTIGACIÓN APLICADA*. San José: Revista Educación.

Velichkov, P. (01 de 08 de 2007). *Designer and Web Architect*. Recuperado el 27 de 03 de 2014, de Designer and Web Architect: <http://blog.creonfx.com/>

Villalba, T. F. (2012). *Modelos de Calidad Del Software*. Madrid: Académica Española.

W3C. (2014). Recuperado el 22 de 02 de 2014, de W3C: <http://www.w3.org/>

Weitzenfeld, A. (2005). *Ingeniería de Software Orientada a Objetos con UML, Java e Internet*. Mexico: Thomson.