

UNIVERSIDAD CATOLICA DE SANTIAGO DEL ESTERO
FACULTAD DE MATEMATICA APLICADA

ESPECIALIZACIÓN EN INGENIERÍA WEB

TRABAJO FINAL INTEGRADOR:

**“Modelado y Desarrollo de un Prototipo de
Aplicación Web con Atributos de Accesibilidad.
Caso de Estudio: Gestión de Guardias Médicas del
Gobierno de Catamarca”**

AUTOR: Lic. Cecilia Elizabeth Gallardo

PROFESOR TUTOR: Dr. Hernán César Ahumada

– 2016 –

AGRADECIMIENTOS

A mi Director, el Dr. Hernán Ahumada por haber dedicado su tiempo con empeño y responsabilidad para guiarme en este trabajo.

A Claudio Andreatta y Oscar Quinteros por ayudarme y salvarme de varias encrucijadas en el mundo de la programación.

Al grupo de personas maravillosas con el que compartí hermosos viajes y el cursado esta Carrera: Nelson Contreras, Marta Miranda, Rafael Garau y Leila Cura.

DEDICATORIAS

Dedico este esfuerzo a mi hermosa familia, a mis hijitas Micaela y Valentina, a mi compañero de vida Oscar, quienes me apoyan en todo momento incondicionalmente.

A mi madre Norma, que siempre está presente alentándome a seguir adelante.

A mi abuelita Ilda.

RESUMEN

En el marco del presente trabajo se plantea el modelado y desarrollo de un prototipo de Aplicación Web para dar solución a la gestión de Guardias Médicas del Gobierno de Catamarca, donde se detecta la necesidad de agilizar y administrar los registros de guardias médicas en forma independiente por cada Área, mediante el acceso vía internet independientemente de la ubicación geográfica de las mismas.

Se adoptó el enfoque UWE para cubrir el ciclo de vida del desarrollo de la WebApp propuesta, teniendo en cuenta además, los principios y enfoques propios de la Ingeniería Web, con el fin de obtener un sistema web de alta calidad, que satisfaga con éxito los requerimientos establecidos.

De acuerdo a las características de los usuarios que operarán con el sistema web desarrollado, fue necesario considerar aquellos atributos de accesibilidad web que asistan a los mismos en el momento del ingreso de datos, para prevenir la introducción de errores y facilitar una interacción con el software de manera eficiente y correcta, independientemente de sus capacidades. Por ello, durante el desarrollo de la presente WebApp se aplicaron las Pautas de Accesibilidad al Contenido Web (WCAG 2.0) correspondientes al Principio “Comprensible” de las mismas, de tal manera que se provea información y manejo de interfaces de usuario comprensibles y accesibles.

A su vez, para evaluar el cumplimiento de dichas características de accesibilidad web, se aplicaron los lineamientos de la estrategia integrada de Medición y Evaluación GOCAME para construir un modelo de conceptos de calidad específico y diseñar las métricas e indicadores en base al mismo, que permitieron medir y evaluar si el prototipo de Aplicación Web terminado satisface las pautas WCAG 2.0 deseadas.

INDICE GENERAL

CAPITULO I - INTRODUCCION.....	12
1.1. INTRODUCCIÓN GENERAL.....	12
1.2. PLANTEAMIENTO DEL PROBLEMA	14
1.3. OBJETIVOS Y ALCANCE	16
1.3.1. <i>Objetivos Principales</i>	16
1.3.2. <i>Alcance</i>	17
1.4. REVISION DE TRABAJOS PREVIOS.....	18
1.4.1. <i>Sistemas software de Administración de Guardias Médicas</i>	18
1.4.2. <i>Guías para alcanzar la Accesibilidad Web</i>	19
1.4.3. <i>Métodos y herramientas para evaluar la Accesibilidad Web</i>	20
1.4.4. <i>Modelos de conceptualización de la calidad del software</i>	23
1.5. RESULTADOS ESPERADOS	24
1.6. JUSTIFICACION DE LA INVESTIGACION	24
CAPITULO II - MARCO REFERENCIAL	28
2.1. APLICACIONES WEB	28
2.1.1. <i>Categoría de Aplicaciones Web</i>	29
2.1.2. <i>Atributos propios de las Aplicaciones Web</i>	31
2.2. INGENIERIA WEB.....	34
2.2.1. <i>Actividades de la Ingeniería Web</i>	36
2.2.2. <i>Características de Aplicaciones Web bien diseñadas</i>	37
2.3. MODELADO DE APLICACIONES WEB	38
2.3.1. <i>Revisión de Métodos de Modelado</i>	40
2.3.2. <i>Lenguaje de Modelado UML</i>	42
2.4. METODOLOGIA UML-BASED WEB ENGINEERING (UWE).....	43
2.4.1. <i>Modelos de UWE</i>	44
2.5. ACCESIBILIDAD WEB	56
2.5.1. <i>Pautas de la Iniciativa de Accesibilidad Web (WAI)</i>	57
2.5.2. <i>Pautas de Accesibilidad al Contenido Web (WCAG)</i>	58
2.6. GESTION DE LA CALIDAD EN LA WEB.....	63
2.6.1. <i>Introducción a la Calidad de un Producto Software</i>	63
2.6.2. <i>Modelos de Calidad Web</i>	66
2.6.3. <i>Estrategia de Medición y Evaluación (M&E) de Calidad GOCAME</i>	67
2.7. TECNOLOGIAS DE DESARROLLO.....	71
2.7.1. <i>Entorno de Desarrollo Integrado (IDE) Netbeans</i>	71
2.7.2. <i>Gestor de Base de Datos PostgreSQL</i>	72
2.7.3. <i>Lenguaje de Programación Groovy</i>	72
2.7.4. <i>Frameworks de Desarrollo Web</i>	75
2.7.5. <i>Patrón Modelo-Vista-Controlador</i>	75
2.7.6. <i>Framework de Desarrollo Grails</i>	76

CAPITULO III - DISEÑO DEL MODELO DE ATRIBUTOS DE CALIDAD	81
3.1. INTRODUCCION	81
3.2. (A1) DEFINICION DE REQUERIMIENTOS NO FUNCIONALES	82
3.2.1. <i>Definición de la Necesidad de Información</i>	83
3.2.2. <i>Selección de un Modelo de Conceptos</i>	83
3.3. (A2) DISEÑO DE LA MEDICIÓN.....	96
3.4. (A4) DISEÑO DE LA EVALUACIÓN	100
3.4.1. <i>Identificación de Indicadores elementales</i>	100
3.4.2. <i>Identificación de Indicadores globales</i>	101
CAPITULO IV - MODELADO Y DISEÑO DEL PROTOTIPO DE APLICACIÓN WEB .	103
4.1. INTRODUCCION	103
4.2. ESPECIFICACION DE REQUISITOS DE SOFTWARE	103
4.2.1. <i>Visión General del Documento</i>	103
4.2.2. <i>Descripción General de la Aplicación Web</i>	108
4.2.3. <i>Requerimientos Específicos</i>	112
4.2.4. <i>Glosario</i>	116
4.3. MODELADO DEL PROTOTIPO DE APLICACIÓN WEB	117
4.3.1. <i>Modelo de Requerimientos (Casos de Uso)</i>	117
4.3.2. <i>Modelo de Contenido</i>	125
4.3.3. <i>Modelo de Navegación</i>	126
4.3.4. <i>Modelo de Presentación</i>	129
CAPITULO V – PRESENTACION DEL PROTOTIPO SIWEBGM	135
5.1. INTRODUCCION	135
5.1.1. <i>Alcance</i>	135
5.1.2. <i>Siglas y Abreviaturas</i>	135
5.2. ESTRUCTURA GENERAL DE LA WEBAPP	136
5.2.1. <i>Esquema General de la Aplicación</i>	136
5.2.2. <i>Imágenes y Botones gráficos</i>	136
5.3. INGRESO A LA APLICACIÓN WEB	138
5.4. OPCIONES DEL SISTEMA SEGÚN ROLES DE USUARIOS	139
5.4.1. <i>Opciones de Menú en común para los tres Roles</i>	139
5.4.2. <i>Opciones de Menú específicas del Rol Supervisor</i>	150
CAPITULO VI - EVALUACION DE LA CALIDAD DEL PROTOTIPO WEB	154
6.1. INTRODUCCION	154
6.1.1. <i>Consideraciones Generales sobre el desarrollo del Prototipo Web</i>	155
6.2. EJEMPLOS DEL CUMPLIMIENTO DE ATRIBUTOS DE ACCESIBILIDAD WEB	156
6.3. (A3) IMPLEMENTACIÓN DE LA MEDICIÓN.....	164
6.4. (A5) IMPLEMENTACIÓN DE LA EVALUACION	166
6.5. (A6) ANALISIS Y RECOMENDACION	167

CAPITULO VII – CONCLUSIONES Y TRABAJOS FUTUROS	169
7.1. CONCLUSIONES FINALES	169
7.2. TRABAJOS FUTUROS	171
ANEXOS	173
ANEXO A – ESPECIFICACION DE METRICAS DE CALIDAD	173
ANEXO B – DESCRIPCION DE CASOS DE USO	185
REFERENCIA BIBLIOGRAFICA	191

INDICE DE FIGURAS

CAPITULO II - MARCO REFERENCIAL

Figura 1: Categoría de Aplicaciones Web	29
Figura 2: Ingeniería Web, un campo multidisciplinario	36
Figura 3: Requerimientos de modelado de Aplicaciones Web.....	38
Figura 4: Desarrollo histórico de Métodos de Modelado de Aplicaciones Web	40
Figura 5: Revisión de Métodos de Modelado de Aplicaciones Web	41
Figura 6: Meta-modelo de UWE	45
Figura 7: Modelo de Casos de Uso aplicando UWE	47
Figura 8: Modelo de Contenido para el caso de estudio de películas on-line.....	48
Figura 9: Modelo de Usuario para el caso de estudio de películas on-line	48
Figura 10: Fragmento de la estructura de navegación desde la Clase Películas	50
Figura 11: Integración de procesos de negocio a la navegación	50
Figura 12: Modelo de presentación de la página Películas	52
Figura 13: Modelo de ciclo de vida de la Adaptación	53
Figura 14: Modelado de la clase Regla	55
Figura 15: Patrón de adaptación.....	55
Figura 16: Relación entre componentes y pautas de Accesibilidad Web de WAI	57
Figura 17: Documentos de WCAG 2.0	59
Figura 18: Mapa visual de las WCAG 2.0	61
Figura 19: Calidad en el ciclo de vida (ISO/IEC 9126-1, 2001)	64
Figura 20: Modelo de Calidad de Producto sistema/software (ISO/IEC 25010, 2011)	65
Figura 21: Características del modelo de calidad y vistas del framework 2Q2U v2.0.	67
Figura 22: Principales conceptos y relaciones dentro del marco C-INCAMI.....	70
Figura 23: Puntos de vista funcional y de comportamiento de proceso de GOCAME	70
Figura 24: Arquitectura MVC	76
Figura 25: Stack de componentes en Grails.....	80

CAPITULO III - DISEÑO DEL MODELO DE ATRIBUTOS DE CALIDAD

Figura 26: Secuencia de Actividades desarrolladas en el presente Trabajo Final	82
--	----

CAPITULO IV - MODELADO Y DISEÑO DEL PROTOTIPO DE APLICACIÓN WEB

Figura 27: Interrelación de SiWebGM con otros sistemas software	108
Figura 28: Actores del Prototipo SiWebGM.....	118
Figura 29: Diagrama de C.U. para el paquete “Acceso al Sistema”	119
Figura 30: Diagrama de C.U. para el paquete “Administrar Guardias Médicas”	119
Figura 31: Diagrama de C.U. para el paquete “Administrar Parámetros”	120
Figura 32: Diagrama de Contenido de SiWebGM	126
Figura 33: Diagrama de Navegación para el Rol AGM.....	128

Figura 34: Fragmento de Diagrama de Presentación de página principal de la WebApp y login de usuario.....	131
Figura 35: Fragmento de Diagrama de Presentación de menú principal y menú de usuario para el actor AGM.....	131
Figura 36: Fragmento de Diagrama de Presentación de componentes de IU involucrados en la Gestión de Guardias Médicas.....	132
Figura 37: Fragmento de Diagrama de Presentación de componentes de IU involucrados en la Gestión de Periodos de Liquidación de Área.....	133
Figura 38: Fragmento de Diagrama de Presentación de componentes de IU involucrados en la Gestión de Servicios Internos de Area.....	133
Figura 39: Fragmento de Diagrama de Presentación de componentes de IU para la página web de emisión de reportes varios.....	134

CAPITULO V – PRESENTACION DEL PROTOTIPO SIWEBGM

Figura 40: Esquema general de la WebApp.....	136
Figura 41: Página principal y formulario de identificación de usuario	139
Figura 42: Opciones de Menú para los usuarios AGM de SAF y de Área	140
Figura 43: Primera instancia de la Gestión de Guardias Médicas	141
Figura 44: Segunda instancia de la Gestión de Guardias Médicas	142
Figura 45: Tercera instancia de la Gestión de Guardias Médicas.....	142
Figura 46: Formulario de modificación o alta de un registro de guardia médica.....	143
Figura 47: Notificación de errores en la página web de modificación o alta de registros de guardias médicas	144
Figura 48: Pagina web de Gestión de Periodos de Liquidación de Área	145
Figura 49: Formulario de creación de registro de Periodos de Liquidación de Area.	146
Figura 50: Página web de Gestión de Servicios Internos de Área	146
Figura 51: Formulario de creación de Servicio Internos de Área	147
Figura 52: Vista del Reporte “Detalle de Guardias por Periodo Liq. Área”	148
Figura 53: Vista del Reporte “Resumen de Guardias por Periodo Liq. Área”	148
Figura 54: Selección del tipo de reporte “Guardias por Agente y Periodo Liq. Area”	149
Figura 55: Vista del Reporte “Guardias por Agente y Periodo Liq. Area”	149
Figura 56: Opciones de menú para el usuario con Rol Supervisor	150
Figura 57: Página web de Gestión de días feriados	151
Figura 58: Formulario de creación de registro de día feriado	151
Figura 59: Página web de Gestión de Duración de Guardias.....	152
Figura 60: Formulario de creación de registros de Duración de Guardias	152
Figura 61: Formulario de creación de usuario de la WebApp	153
Figura 62: Página Web de Gestión de Usuarios de la WebApp	153

CAPITULO VI - EVALUACION DE LA CALIDAD DEL PROTOTIPO WEB

Figura 63: Página web de Generación de Reportes Varios.....	158
Figura 64: Página web de Generación de Reportes Varios luego de iniciado cambio de contexto	159
Figura 65: Notificación textual de error de campo obligatorio de navegador web ..	159
Figura 66: Notificación textual de error de campo obligatorio desde la WebApp....	160
Figura 67: Ejemplo de definición de formato de atributo en una clase de dominio.	161
Figura 68: Notificación textual de error para campo con determinado formato.....	161
Figura 69: Identificación de campos con un determinado formato	162
Figura 70: Ejemplo de identificación de campos obligatorios	163
Figura 71: Ejemplo de asociación de etiquetas con controles de formularios	163
Figura 72: Ejemplo de ubicación de etiquetas de campo de texto, lista desplegable y casilla de verificación	164

INDICE DE TABLAS

CAPITULO II - MARCO REFERENCIAL

Tabla 1: Tecnologías software de desarrollo Web.....	71
--	----

CAPITULO III - DISEÑO DEL MODELO DE ATRIBUTOS DE CALIDAD

Tabla 2: Árbol de requerimientos para evaluar la “Accesibilidad en la comprensión” de una WebApp	85
Tabla 3: Definición de métricas para el Atributo 1.1.1 del modelo de requerimientos	98
Tabla 4: Definición de métricas para el Atributo 1.3.2.2 del modelo de requerimientos .	99
Tabla 5: Definición de Indicador Elemental	101

CAPITULO IV - MODELADO Y DISEÑO DEL PROTOTIPO DE APLICACIÓN WEB

Tabla 6: Descripción del Caso de Uso: Gestionar Guardias Médicas.....	121
Tabla 7: Descripción del Caso de Uso: Seleccionar Periodo de Liquidación de Área	123
Tabla 8: Descripción del Caso de Uso: Seleccionar Contrato de Guardia.....	124
Tabla 9: Descripción del Caso de Uso: Emitir Informes Varios	124

CAPITULO VI - EVALUACION DE LA CALIDAD DEL PROTOTIPO WEB

Tabla 10: Resultados de la medición de atributos de calidad	165
Tabla 11: Resultados de la evaluación de calidad del prototipo de aplicación web	166

CAPITULO I - INTRODUCCION

1.1. INTRODUCCIÓN GENERAL

Las Aplicaciones Web modernas representan sistemas de software complejos, en los cuales interactúan diversas tecnologías y estándares, por lo tanto, el desarrollo de dichas aplicaciones requiere la utilización de un enfoque de ingeniería metodológicamente sólido.

Basada en la Ingeniería de Software, la *Ingeniería Web* comprende el uso de principios científicos, de ingeniería y de enfoques disciplinados y sistemáticos para la especificación, implementación, operación y mantenimiento de Aplicaciones Web de alta calidad, para satisfacer con éxito la diversidad de usuarios y los requerimientos únicos de este tipo de sistemas (Murugesan, Deshpande, Hansen, & Ginige, 1999). La Ingeniería Web no representa una sola actividad sino que consiste en un proceso, por lo que abarca una serie de áreas como ser: ingeniería de requisitos; modelado de sistemas Web; arquitectura web; diseño de páginas web; interacción con bases de datos; calidad web; usabilidad web; seguridad Web; pruebas Web; metodologías de desarrollo Web; métricas Web; gestión de proyectos Web, entre otros (Murugesan, 2008).

En particular, uno de los requerimientos de calidad a considerar en el desarrollo de aplicaciones web es la *Accesibilidad Web*, donde el Consorcio de la World Wide Web¹ (W3C), expresa al respecto que todas las personas, principalmente aquellas con discapacidad, puedan percibir, entender, navegar, interactuar con la Web, y también poder contribuir a la misma.

Entre el conjunto de los componentes del desarrollo Web que deben interactuar para que la Web sea accesible se encuentra el “contenido

¹ Sitio Web de W3C: <https://www.w3.org/>

web”, el cual se refiere a la información dentro de una página o Aplicación Web incluyendo *información natural*: como texto, imágenes y sonidos o *código o marcado*: que define la estructura, presentación, etc.

Para ayudar a hacer el contenido web más accesible, la Iniciativa de Accesibilidad Web² (WAI) desarrolló las “Pautas de Accesibilidad para el Contenido Web” (Web Content Accessibility Guidelines 2.0, 2008), las cuales hoy constituyen un estándar internacional y proporcionan pautas, criterios de éxito comprobables y técnicas, los cuales se utilizan para evaluar los requerimientos de accesibilidad web de acuerdo a las necesidades de diferentes grupos y situaciones.

Asimismo, se considera también a la “accesibilidad” como una sub-característica de la “usabilidad” según el modelo de Calidad de Producto Software propuesto por el estándar ISO/IEC 25010 (2011) el cual define el conjunto de características y relaciones entre ellas para proveer las bases para especificar los requerimientos de calidad y su evaluación.

Para el desarrollo de este trabajo se aplicaron los procesos, metodologías y técnicas propias de la Ingeniería Web, como así también los requerimientos y la complejidad particular que implica el desarrollo de una Aplicación Web con atributos de accesibilidad, para dar solución al problema que se plantea.

Cabe destacar, que en el marco del presente Trabajo Final, se han realizado las siguientes publicaciones en dos prestigiosos Congresos:

- ✎ **Evento:** CAIS 2015 - 6º Congreso Argentino de Informática y Salud
- Nombre Trabajo:** Modelado de un Prototipo de Aplicación Web para la Gestión de Guardias Médicas del Gobierno de Catamarca
- Fecha evento:** Agosto de 2015
- Ciudad:** Rosario, Santa Fe
- Título publicación congreso:** Contribuciones CAIS 2015

² Sitio Web de WAI: <https://www.w3.org/WAI/>

ISSN: 2451-7607

URL: <http://44jaiio.sadio.org.ar/sites/default/files/cais47-66.pdf>

Institución organizadora: Universidad Nacional de Rosario

↪ **Evento:** Il Congreso Argentino de Ingeniería CADI

Nombre Trabajo: Análisis del grado de cumplimiento de Pautas de Accesibilidad al Contenido Web a través de la medición y evaluación de Calidad

Fecha evento: Septiembre de 2014

Ciudad: San Miguel de Tucumán

Título publicación congreso: Segundo Congreso Argentino de Ingeniería

ISBN: 978-987-1662-51-7

Institución organizadora: Universidad del Norte Santo Tomas de Aquino. UNSTA

1.2. PLANTEAMIENTO DEL PROBLEMA

En el contexto de este trabajo se identificaron tres tipos de problemas o necesidades que a continuación se detallan.

La primera necesidad se manifiesta dentro del dominio del “Sistema Administrativo de Guardias” del Gobierno de la Provincia de Catamarca, donde se celebran contratos especiales de guardia entre personal profesional o técnico y un Área Programática en particular, para prestar servicios médicos, de mantenimiento y otros que se requieran ser cubiertos para garantizar la continuidad del normal funcionamiento de los Establecimientos Sanitarios de toda la provincia.

El principal inconveniente que se detecta en este Sistema Administrativo de Guardias, es la demora que se genera mensualmente en la liquidación

y pago de las guardias efectivamente realizadas, debido a que la gestión y control de la información de los registros de guardias de todos las Áreas se centraliza físicamente en una sola dependencia del Gobierno en la capital provincial.

Por ello es preciso agilizar, descentralizar y delegar la responsabilidad de la gestión de los registros de guardias del “Sistema Administrativo de Guardias” al Establecimiento Sanitario que corresponda, los cuales se encuentran distribuidos geográficamente en toda la provincia. A su vez, es necesario que toda esta información se encuentre integrada e interconectada para garantizar una mayor consistencia de la misma y así también debe estar disponible para realizar los controles y auditorias pertinentes por parte de las Autoridades que la requieran.

Otro problema que se plantea en este trabajo se relaciona con las necesidades de los posibles usuarios que tendrá la Aplicación Web. Se observa que los empleados administrativos que están involucrados en el Sistema Administrativo de Guardias, sobre todo del interior de la Provincia, en algunos casos son personas de edad avanzada, que generalmente tienen escasa experiencia de interacción con computadoras y sistemas software, y que además tienen escaso conocimiento del dominio del problema de guardias y de las inconsistencias que pueden surgir. Esto genera muchos inconvenientes y una alta probabilidad en la introducción de errores y confusiones si no se brindan contenidos, instrucciones e interfaces comprensibles y accesibles al momento del registro de la información mediante el software.

Por ello es indispensable considerar las características de calidad respecto a la *accesibilidad del contenido web* que debe incluir el software que se desarrolle para el gobierno provincial. En el contexto del Sistema de Guardias, es relevante considerar aquellos atributos de accesibilidad web que asistan al usuario del Sistema Administrativo de Guardias en el

momento del ingreso de datos, proporcionando información y manejo de interfaces de usuario *compresibles y accesibles* para prevenir la introducción de errores y facilitar una interacción con el software de manera eficiente y correcta, independientemente de sus capacidades.

Por otra parte, el último problema identificado tiene que ver con la evaluación de las características de accesibilidad web que el prototipo desarrollado satisface. Dicha evaluación se puede realizar mediante herramientas software automatizadas o en forma manual. La evaluación de la accesibilidad web realizada mediante herramientas software automatizadas permite realizar una revisión rápida, ayuda a tener una primera impresión de la accesibilidad de una página o aplicación web, pero no proporciona un análisis definitivo y fiable, ya que pueden no detectar errores importantes o suele señalar errores que realmente no existen (falsos positivos).

Considerando además que un concepto de calidad de alto nivel como la Accesibilidad, es una compleja combinación de factores que se deben evaluar en la práctica por medio de la cuantificación de atributos de entidades de abstracción de más bajo nivel y que algunos de estos atributos requieren de un juicio humano para su medición y evaluación, es necesario definir un mecanismo que permita medir y evaluar los requisitos de accesibilidad web que se determinen que la Aplicación Web del dominio de guardias debe contemplar.

1.3. OBJETIVOS Y ALCANCE

1.3.1. Objetivos Principales

De manera sintética, a continuación se exponen los objetivos propuestos para solucionar los problemas planteados:

- Modelar y desarrollar un prototipo de Aplicación Web aplicando pautas de accesibilidad web de WCAG 2.0, que permita gestionar en forma independiente por cada Área Programática, los registros de las guardias médicas realizadas por el personal contratado por el gobierno de la provincia de Catamarca y que provea interfaces e información comprensibles a los usuarios.
- Construir un modelo de conceptos de calidad y diseñar las métricas e indicadores en base al mismo, que permitan medir y evaluar si el prototipo de Aplicación Web terminado satisface las pautas WCAG 2.0 establecidas.

1.3.2. Alcance

El presente proyecto contempla los siguientes ítems:

- Identificación y aplicación de las directivas y técnicas de WCAG 2.0 para cumplir con el Principio “Comprensible” Nivel A de WCAG 2.0, para proporcionar información y manejo de interfaces de usuario comprensibles en el prototipo de Aplicación Web.
- Durante las etapas de modelado y diseño del prototipo de la aplicación web, se consideran las pautas y criterios del Principio “Comprensible” Nivel A de WCAG 2.0, con el fin de especificar su intervención en los modelos realizados.
- El prototipo de aplicación web para la gestión de guardias médicas abarca las siguientes funcionalidades:
 - ✓ Gestión de usuarios del sistema
 - ✓ Gestión de los Servicios Internos que ofrece cada Área Programática
 - ✓ Gestión de días no laborables
 - ✓ Gestión de Periodos de Liquidación

- ✓ Emisión de reportes varios
- ✓ Gestión de los registros de guardias médicas realizadas por el personal

1.4. REVISION DE TRABAJOS PREVIOS

1.4.1. Sistemas software de Administración de Guardias Médicas

Desde el punto de vista académico, existen trabajos relacionados con el dominio del problema de la planificación de las guardias médicas en hospitales. En (Aviles Monroy, Navarro Ramirez, Toapaxi Acosta, & Echeverria Briones, 2015) y (Gómez, Puente, Priore, Pino, & Fernández, 2007) se utilizan técnicas de minería de datos para intentar predecir y confeccionar automáticamente las asignaciones de personal para atender las urgencias médicas que surjan en los hospitales, en base a datos de los pacientes y patologías que se hayan tratado en el área de urgencias, de acuerdo a la experiencia del personal y otras restricciones específicas del área. El prototipo de aplicación web desarrollado en el presente trabajo permite el registro de las guardias médicas efectivamente realizadas por el personal, por lo que los objetivos de los documentos antes mencionados quedan fuera del alcance de este trabajo.

Por otra parte, se encontraron referencias sobre desarrollos de sistemas de información para la gestión de guardias médicas realizadas por personal dentro de hospitales o establecimientos asistenciales (GuardiaGes, 2015) (SICATU, 2015). En general, la funcionalidad de estos sistemas, es muy similar a la planteada en la presente propuesta, pero con la diferencia que no se tratan de aplicaciones que funcionen en una plataforma Web, donde el objetivo principal sea la descentralización de la administración de las guardias al establecimiento asistencial que corresponda, y además, no se puede determinar la utilización de

metodologías y herramientas propias de la ingeniería web y de software. Asimismo, no se verifican ni especifican en estos sistemas el cumplimiento de requisitos de calidad en cuanto a la accesibilidad web. Por lo tanto, se concluye que la presente se diferencia en gran medida de los desarrollos existentes, aportando un prototipo de aplicación web que fue desarrollado siguiendo las prácticas y procesos propios de la ingeniería web y que contempla atributos de accesibilidad web respecto a la entrega de información e interfaces de usuarios comprensibles.

1.4.2. Guías para alcanzar la Accesibilidad Web

En lo que se refiere a la accesibilidad web, el Consorcio W3C es uno de los principales referentes y que ha trabajado durante más de diez años, a través de la Iniciativa de Accesibilidad Web (WAI), en el desarrollo de un estándar denominado Pautas de Accesibilidad para el Contenido Web, el cual es considerado un punto de referencia para la mayoría de la legislación en materia de Tecnologías de la Información y Comunicación en todo el mundo. El WCAG tiene dos versiones de documentos: WCAG 1.0 (Web Content Accessibility Guidelines 1.0, 1999) y WCAG 2.0 (Web Content Accessibility Guidelines 2.0, 2008).

En Argentina, existe una regulación en cuanto a la Accesibilidad Web mediante la Ley Nacional N° 26.653 denominada "Ley de Accesibilidad de la Información en las Páginas Web", donde se establece que todos los Organismos dependientes del Estado Nacional *"deberán respetar en los diseños de sus páginas Web las normas y requisitos sobre accesibilidad de la información que faciliten el acceso a sus contenidos, a todas las personas con discapacidad con el objeto de garantizarles la igualdad real de oportunidades y trato, evitando así todo tipo de discriminación"*. A su vez, el 27 de junio de 2011 a través de la Resolución N° 69/2011 de la Secretaría de Gabinete de la Nación se aprobó el documento "Guía de Accesibilidad para Sitios Web del Sector Público Nacional", la cual ha sido

pensada para todos los desarrolladores de contenidos de la Web y para los desarrolladores de herramientas de creación en el ámbito del Sector Público Nacional. Esta guía se ajusta a los lineamientos de las WCAG versión 1.0.

Otro hito importante es que en agosto de 2011, Argentina se convirtió en un miembro de la W3C.

En la provincia de Catamarca, no existe una legislación que se refiera a la accesibilidad web, por lo que en este trabajo, continuando el camino marcado por la legislación nacional, se considerarán los criterios y directivas de la última versión estable de las Pautas de Accesibilidad al Contenido Web de WAI-W3C: las WCAG 2.0.

Las WCAG 2.0 proporcionan 12 pautas que se organizan bajo 4 principios, donde para cada pauta se proveen criterios de éxito comprobables, los cuales se utilizan para evaluar los requerimientos de accesibilidad web de acuerdo a las necesidades de diferentes grupos y situaciones, definiendo tres niveles de conformidad: A (más bajo), AA, y AAA (más alto). El objetivo final es detectar puntos no cumplidos para que las páginas involucradas sean redefinidas.

1.4.3. Métodos y herramientas para evaluar la Accesibilidad Web

Actualmente existen herramientas software de evaluación de la accesibilidad Web que ayudan a determinar si un sitio o aplicación Web cumple con las normas de accesibilidad de WCAG en los contenidos de una página web. Estas herramientas pueden reducir significativamente el tiempo y esfuerzo de la evaluación, pero ninguna puede determinar totalmente en forma automática la accesibilidad de los sitios Web, ya que algunos criterios comprobables requieren de un análisis subjetivo y manual por parte del evaluador debido a la complejidad y comprensión que implica el contenido de una página web.

Como referencia, se pueden mencionar herramientas software de evaluación de accesibilidad Web tales como: t.a.w.³, WAVE⁴ (Web Accessibility evaluation tool), eXaminator⁵, entre otras mencionadas en la lista de herramientas para la accesibilidad web dadas por la WAI (Web Accessibility Evaluation Tools List, 2016).

Estas herramientas software son muy útiles porque soportan una amplia variedad de controles de accesibilidad, pero ninguna puede analizar automática y completamente todos los criterios éxito de WCAG 2.0, ya que algunos de ellos requieren obligatoriamente de una revisión humana.

Por otra parte, la medición y evaluación de la accesibilidad también se puede realizar mediante métricas. Una definición del concepto de métrica dada por (ISO/IEC TR 9126-2, 2003) es: *“el método de medición definido y la escala de medición”*, y más informalmente se puede definir como: *“cualquier medida destinada a conocer o estimar atributos de calidad de un artefacto”*. Las métricas son importantes para controlar, entender y mejorar productos y procesos en el desarrollo de software.

Las métricas cuantitativas para la accesibilidad web ayudan a sintetizar el resultado de las evaluaciones proporcionando el grado de accesibilidad cumplido. El objetivo de estas métricas es tener una noción acerca de cuán accesible es el artefacto evaluado.

Existen varios trabajos donde se proponen distintos enfoques de métricas e indicadores para medir y evaluar la accesibilidad desde diferentes puntos de vista.

La Métrica Cuantitativa para la Accesibilidad Web (WAQM), propuesta por (Vigo, Arrue, Brajnik, Lomuscio, & Abascal, 2007), se calcula automáticamente a partir de los reportes de dos herramientas software de

³ Sitio Web de t.a.w.: <http://www.tawdis.net/> - Fundación Centro Tecnológico de Inf. y Comunicación

⁴ Sitio Web de WAVE: <http://wave.webaim.org/> - Web Accessibility In Mind (WebAIM)

⁵ Sitio Web de eXaminator: <http://examinator.ws/> - Desarrollado por Carlos Benavidez

evaluación automática: EvalAccess y LIFT. Los reportes de evaluación se basan en la norma WCAG 1.0 pero WAQM también proporciona un valor de accesibilidad para los puntos de comprobación de WCAG 2.0, basado en una tabla de correspondencia de los puntos de control de las dos versiones de WCAG.

Como WAQM se basa en los informes producidos por herramientas automáticas, los puntos de comprobación que pueden ser evaluados de forma automática tienen una mayor influencia en los resultados finales que los problemas semi-automáticos. Lo cual no sería totalmente correcto si se considera que algunos puntos de comprobación que requieren de un juicio humano, tienen igual o mayor importancia que aquellos que se pueden analizar en forma objetiva u automática.

Por otra parte, en (Sullivan & Matson, 2000) se evalúan 8 puntos de control de WCAG 1.0. Como resultado, se obtiene una tasa de fallos (failure-rate) que representa la proporción entre los posibles puntos de fallo y los errores reales comprobados. Por lo tanto, el rango de resultado varía entre 0 y 1. Con esta métrica, algunos factores como el impacto del error, la naturaleza del error y otros requerimientos no son tomados en cuenta. Por lo tanto, este enfoque es adecuado para medir cuantitativamente la accesibilidad con respecto a la conformidad (el cumplimiento requiere una tasa de fracaso igual a 0), pero plantea algunas inconsistencias para medir la accesibilidad, ya que más barreras de accesibilidad implican menos accesibilidad, situación que este indicador no refleja de manera correcta.

Otro trabajo propuesto es la fórmula WAB (Web Accessibility Barrier) (Hackett, Parmanto, & Zeng, 2004), la cual utiliza como parámetros de entrada el total de páginas de un sitio web, los errores totales de accesibilidad, así como los posibles errores en una página web y la prioridad de los mismos. Estos valores se computan para 25 puntos de

control de WCAG 1.0 encontrados en una página. Las puntuaciones devueltas por este método no se limitan a un rango limitado de valores, por lo que dicho enfoque puede resultar útil solo para una puntuación de páginas web acorde a su nivel de accesibilidad. El principal inconveniente de esta métrica es que teniendo en cuenta solo el resultado de una página web única, no es posible tener una referencia de la accesibilidad ya que no hay límites para buenos o malos niveles de accesibilidad.

En (Vigo & Brajnik, 2011) se realiza una comparación y evaluación de varias propuestas de métricas automáticas y semiautomáticas para evaluar la accesibilidad web.

Se puede apreciar en este último trabajo mencionado, que no existe ningún método que contemple totalmente los puntos de comprobación de las pautas WCAG 2.0, ya que la mayoría de los enfoques se basan en pruebas automáticas de accesibilidad, dejando de lado aquellos que requieren un juicio humano (o asignando a estos errores una importancia menor).

1.4.4. Modelos de conceptualización de la calidad del software

Desde el punto de vista de la Ingeniería del software y Web, además de los modelos de calidad de software (ISO/IEC 25010, 2011) y (ISO/IEC 9126-1, 2001), mencionados anteriormente, también se pueden referenciar los propuestos por (Boehm, Brown, Kaspar, & Lipow, 1978), (McCall, Richards, & Walters, 1977) y (Mich, Franch, & Gaio, 2003), entre otros. El objetivo principal de dichos modelos es identificar los requisitos o factores que contribuyen a la calidad del software.

Sin embargo, se observa que no existe un modelo de calidad de referencia que identifique y defina, de manera exhaustiva, los atributos que permitan medir y evaluar un concepto de alto nivel y tan complejo como la Accesibilidad.

En conclusión, se propone en este trabajo, construir un modelo de calidad específico que permita contemplar, medir y evaluar todos los criterios de éxito comprobables para cumplir con las pautas de accesibilidad web planteadas en este trabajo. Para completar la evaluación, se plantea el diseño de las métricas e indicadores correspondientes.

1.5. RESULTADOS ESPERADOS

El resultado esperado de la presente propuesta, es un prototipo de Aplicación Web que permita gestionar los registros de guardias médicas realizadas por el personal del gobierno provincial, que cumpla con los criterios de éxito del Principio **Comprensible** Nivel A de las pautas de accesibilidad al contenido web (WCAG 2.0), de manera que todos los usuarios puedan interactuar y entender el contenido y las interfaces de usuario de la aplicación web sin inconvenientes y de una manera más eficiente. Para evaluar que el prototipo creado cumple con las pautas de accesibilidad establecidas, se realizará la medición y evaluación de dichos atributos de calidad, mediante el desarrollo de un modelo de conceptos específico y el diseño de métricas e indicadores pertinentes.

1.6. JUSTIFICACION DE LA INVESTIGACION

Uno de los propósitos principales de este trabajo es aplicar los principios de la Ingeniería Web en cada etapa de la construcción del prototipo de la aplicación web propuesto, utilizando una metodología de desarrollo web para el análisis, modelado y diseño del mismo, considerando requerimientos de calidad web como la accesibilidad y evaluando posteriormente el cumplimiento de la misma mediante la construcción de un modelo de calidad específico junto con el diseño de métricas web e indicadores para tal fin.

Por otra parte, uno de los problemas al que intenta brindar solución el prototipo de Aplicación Web propuesto es la demora en la liquidación y pago de guardias médicas a raíz de una administración ineficiente de las mismas, ya que se genera un cuello de botella en la carga y control por encontrarse centralizada esta tarea en un solo Organismo.

Mediante el prototipo de aplicación web propuesto en este trabajo, la administración de los registros de guardias médicas podrá realizarse desde cada Área o Establecimiento Sanitario donde se originan los mismos, emitiendo desde la Aplicación Web, todas las planillas necesarias para conformar los expedientes de pago mensuales de guardias en forma independiente por Área. Esta aplicación web debe integrarse al sistema actual, el cual continuará gestionando los contratos del personal, los establecimientos sanitarios, entre otros parámetros.

Con esta nueva propuesta de gestión de las guardias, se reducen en su mayoría errores e inconsistencias de la información, aumentando considerablemente la confiabilidad de la misma y la agilidad en el registro de datos. Esto permite disminuir los tiempos necesarios para la liquidación y pago de guardias, lo cual beneficia por una parte, a la gestión de Organismos de control y supervisión del Estado Provincial, quienes podrán tener acceso en línea a la información de guardias y dejarán de realizar verificaciones en forma manual y por otra parte, los principales beneficiarios de esta nueva propuesta serán los propios empleados que realizan las guardias médicas, percibiendo los haberes correspondientes, en tiempo y forma.

Al considerar que la aplicación a desarrollar deba funcionar en una plataforma web, facilita la actualización y mantenimiento de la misma, al no tener la necesidad de distribuir e instalar software en cada computadora cliente de los Establecimientos distribuidas geográficamente en toda la provincia, sino solamente actualizando el sistema desde el

Servidor Web. Por ello también se verá beneficiado el personal encargado del desarrollo y mantenimiento de la Aplicación Web.

Otro aspecto importante de este trabajo, es el hecho de considerar que el prototipo de aplicación web desarrollado deba cumplir con requisitos de calidad en cuanto a la accesibilidad web, respecto al Principio Compresible de las WCAG 2.0, lo cual beneficiará a los propios usuarios de la Aplicación Web permitiéndoles navegar, entender e interactuar con la misma de manera correcta, sin ambigüedades ni confusiones, independientemente de sus características, capacidades o conocimientos del dominio del problema.

En el presente proyecto, se plantea como objetivo la aplicación y cumplimiento sólo de las pautas referidas al Principio “Comprensible” Nivel A de WCAG 2.0, el cual aborda la problemática de la entrega de contenido y manejo de interfaces de usuario compresibles, mediante la asistencia en el ingreso de datos y en la ayuda para evitar y corregir errores en formularios. Consideramos que estas pautas de accesibilidad web son las más significativas para el tipo de la Aplicación Web a desarrollar, la cual se trata de un tipo transaccional, donde el principal componente de interfaz e interacción con el usuario son los formularios web para el ingreso de datos. Además, generalmente en este tipo de aplicaciones no se utilizan imágenes ni medios tempo-dependientes que transmitan algún tipo de información y tampoco la estética de la misma toma relevancia. Por otra parte, la aplicación web propuesta, será operada por usuarios que son empleados en actividad del gobierno desde equipos de computadora instalados en oficinas. Es por ello que en general, las demás pautas y principios no afectan a la accesibilidad del tipo de aplicación, contexto de uso y usuarios de la aplicación propuesta, no así el caso de las pautas del principio comprensible de WCGA 2.0, que benefician tanto a los usuarios que pudieran tener algún tipo de

discapacidad leve, o que tengan poca experiencia con el manejo de sistemas software, a gente mayor y al resto de personas en general.

Por otro lado, al incluir y guiar desde el inicio del proceso de desarrollo del prototipo a través de las características y pautas de accesibilidad web que debe cumplir la aplicación web final, se obtiene una mayor probabilidad de que el producto terminado incluya los requisitos de calidad, evitando tareas de mantenimiento y re-diseño para incorporar los requisitos que resultaran insatisfechos en la etapa de evaluación de la aplicación terminada.

Por último, se propone en este trabajo, construir un modelo de calidad y diseño de métricas e indicadores específicos conforme a los requerimientos de accesibilidad web del principio Comprensible de WCAG 2.0, por lo que este aporte facilita al evaluador una manera consistente y uniforme para la medición y evaluación de dichos atributos de calidad, muchos de los cuales requieren de una revisión humana por lo que no se pueden utilizar herramientas de evaluación automáticas en estos casos.

CAPITULO II - MARCO REFERENCIAL

2.1. APLICACIONES WEB

En los comienzos de la era web (alrededor de 1990 a 1995), los “sitios web” consistían en poco más que un conjunto de archivos hipertexto vinculados que presentaban información utilizando texto y gráficos limitados (Murugesan, 2008). Con el paso del tiempo, el Lenguaje de Marcado de Hipertexto (HTML) fue ampliado por el desarrollo de tecnologías y herramientas tales como: Lenguaje de Marcado Extensible (XML), Java, etc., lo que permitió a los ingenieros web proporcionar capacidad computacional tanto del lado del cliente como del servidor junto con el contenido. Es aquí donde surgen los sistemas y aplicaciones basados en la web.

Actualmente, las Aplicaciones Web han evolucionado hasta convertirse en herramientas informáticas sofisticadas que no sólo proporcionan una funcionalidad independiente para el usuario final, sino que también se han integrado con bases de datos y aplicaciones corporativas y gubernamentales.

Una **Aplicación o Sistema Web** podría definirse entonces como *“una aplicación cliente / servidor que utiliza un navegador Web como su programa cliente, y lleva a cabo un servicio interactivo mediante la conexión con servidores Web a través de Internet o Intranet”* (Shklar & Rosen, 2003).

Un sitio Web sólo ofrece contenido de los archivos estáticos. En cambio, una aplicación web presenta contenido adaptado dinámicamente en función de parámetros de la petición, seguido del comportamiento de los usuarios, y las consideraciones de seguridad.

2.1.1. Categoría de Aplicaciones Web

De acuerdo a su cronología de desarrollo y el grado de complejidad, se pueden identificar, entre otras, las siguientes categorías de Aplicaciones Web, tal como se muestra en la Figura 1: (Kappel, Pröll, Reich, & Retschitzegger, 2003)

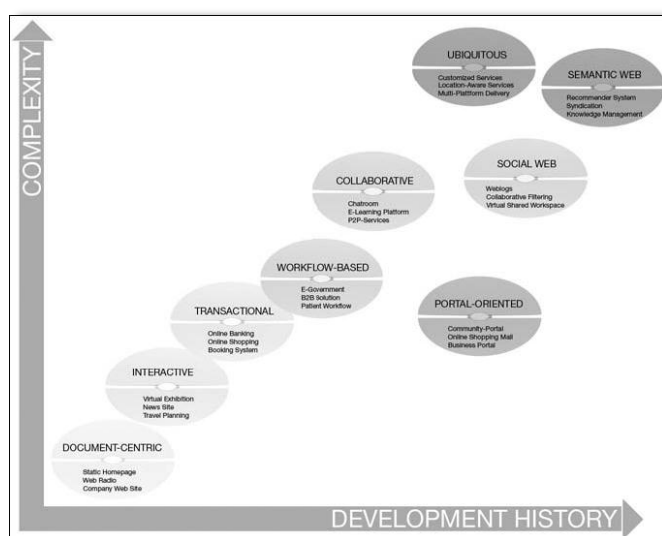


Figura 1: Categoría de Aplicaciones Web

- **Sitios web centrados en documentos:** representan las páginas web estáticas de solo lectura almacenadas en el Servidor Web. Ejemplos: página estática principal de una organización (homepage), libros online, etc.
- **WebApp interactivas:** surgen con la introducción de Common Gateway Interface (CGI) y Formularios HTML. Los enlaces y páginas web se generan dinámicamente según la entrada del usuario. Ejemplos: sitios de noticias, de exposiciones virtuales, formularios de registración, etc.
- **WebApp transaccionales:** proporcionan más interactividad, dando al usuario la posibilidad de realizar cambios en el contenido. Se requiere el uso de sistemas de Bases de Datos que permiten el manejo eficiente

y consistente del contenido y consultas estructuradas. Ejemplos: homebanking, sistemas de compras y reservas on-line.

- **WebApp basadas en workflow:** permiten el manejo de los flujos de trabajo dentro o entre diferentes organizaciones. Esto es posible gracias a la disponibilidad de servicios web para garantizar la interoperabilidad. Ejemplos: soluciones Business-to-Business (B2B) en el comercio electrónico, aplicaciones de gobierno electrónico en el ámbito de la administración pública.
- **WebApp colaborativas:** se emplean para fines de cooperación en operaciones no estructuradas (groupware) y entre usuarios. Apoyan la información y espacios de trabajo compartido con el fin de generar, editar y gestionar la información compartida. Ejemplos: Wiki, Weblogs, plataformas de e-learning.
- **Web social:** las personas ofrecen su identidad a una comunidad (pequeña) de otras personas con intereses similares con el propósito de buscar objetos relacionados de interés. Ejemplos: Weblogs, Facebook.
- **WebApp orientadas a Portal:** proporcionan un único punto de acceso a fuentes de información y servicios independientes y potencialmente heterogéneos de información. Ejemplos: portal general de Yahoo, portales empresariales, portales de negocio en forma de centros comerciales on-line, portales comunitarios, etc.
- **WebApp ubicuas:** ofrecen servicios personalizados en cualquier momento en cualquier lugar y para cualquier dispositivo, facilitando así el acceso en todas partes (ubicuo). Ejemplo: Web que muestra el menú del día en los dispositivos móviles de los usuarios que ingresan un restaurante en cierto horario.
- **Web Semántica:** el objetivo es presentar la información no sólo para seres humanos, sino también en una forma legible por máquinas. Esto

facilitaría la gestión del conocimiento en la Web, la vinculación y la reutilización del conocimiento. Ejemplo: sistemas de recomendación.

En la mayoría de estas categorías existe una correlación entre la cronología de desarrollo y la complejidad. Por ejemplo, las aplicaciones basadas en Workflow son aplicaciones transaccionales, es decir que el nivel más alto de desarrollo requiere el desarrollo previo de una categoría de menor complejidad. Sin embargo, hay excepciones sobre esta regla en algunas categorías donde surgen históricamente desarrollos más recientes pero con un menor grado de complejidad, como es el caso de las aplicaciones web orientadas a portal.

Cada una de estas categorías tiene sus propios campos específicos de aplicación. En consecuencia, las aplicaciones Web más complejas, típicamente se pueden asignar a varias categorías a la vez.

2.1.2. Atributos propios de las Aplicaciones Web

Existe un amplio debate sobre este tema, donde algunas personas argumentan que una Aplicación Web no es más que una aplicación cliente-servidor y que tiene los mismos atributos que una aplicación software convencional. En cambio otros autores, sostienen que cuando se considera en su totalidad el conjunto completo de características de las Aplicaciones Web se diferencian de los sistemas software convencionales.

Las principales características o atributos propios de las Aplicaciones Web son: (Murugesan & Ginige, 2005) (Pressman & Lowe, 2009)

- **Contenido.** El contenido de una aplicación web, que puede incluir texto, gráficos, imágenes, audio y/o vídeo, se integra con el procesamiento procedural. La forma en que se presenta y organiza dicho contenido tiene implicaciones en el rendimiento y el tiempo de

respuesta del sistema. Además, la presentación del contenido debe ser significativa de acuerdo al tipo de usuario de la aplicación.

- **Intensidad de red.** Cada aplicación web reside en una red y debe servir a las necesidades de una comunidad diversa de clientes. La comunicación entre el cliente de la aplicación web y los servidores web se realiza a través de internet.
- **Concurrencia, facilidad de uso.** Las aplicaciones Web están destinadas a ser utilizadas por un amplio número variable de usuarios, con diversos requisitos, expectativas y habilidades. Por lo tanto, las características de la interfaz de usuario y usabilidad tienen que satisfacer las necesidades de una amplia comunidad de usuarios anónimos a quien no podemos ofrecer capacitación, lo que complica la interacción en la Web humana, interfaz de usuario y la presentación de la información.
- **Carga impredecible.** El número de usuarios de una aplicación web puede variar en órdenes de magnitud en el día a día, por lo tanto una WebApp debe ser capaz de manejar un número indeterminado de eventos simultáneamente.
- **Rendimiento.** Las aplicaciones web deben proveer tiempos de esperas cortos y razonables ya sea para el acceso a la misma, como para el procesamiento del lado del servidor, formateo de la salida del lado del cliente, etc. Pueden existir además, aplicaciones web donde el rendimiento sea crítico, ya que la vida humana puede estar en juego.
- **Disponibilidad.** Aunque la expectativa de 100 por ciento de disponibilidad no es razonable, los usuarios de Aplicaciones Web populares a menudo exigen el acceso de forma "24/7/365".
- **Dirigido por datos.** Si bien la función primaria de muchas Aplicaciones Web es utilizar hipermedia para presentar texto, gráficos, audio y contenido de vídeo para el usuario final, la gran mayoría de WebApp se

utilizan comúnmente para acceder a información que existe en Bases de Datos, las cuales no forman parte integral del entorno basado en la Web (por ejemplo, aplicaciones de comercio electrónico o financieras).

- **Evolución continua.** A diferencia de las aplicaciones de software convencional, que evolucionan a lo largo de una serie de versiones cronológicamente espaciadas y planificadas, las Aplicaciones Web evolucionan continuamente. En muchos casos, no es posible especificar completamente lo que un sitio Web debe contener al inicio del proceso de desarrollo, debido a que su estructura y funcionalidad evolucionan con el tiempo, especialmente después de que el sistema se pone en funcionamiento. Gestionar el cambio y la evolución de una aplicación web es uno de los principales retos técnicos, organizacionales y de gestión, mucho más exigente que un desarrollo de software tradicional.
- **Inmediatez.** Se traduce en la necesidad imperiosa de lanzar software al mercado rápidamente, es una característica de muchos dominios de aplicación, y más aún en las Aplicaciones Web que con frecuencia presentan un tiempo de comercialización de unos pocos días o semanas. Los ingenieros Web deben utilizar métodos de planificación, análisis, diseño, implementación y pruebas que se adapten a los tiempos escasos disponibles para el desarrollo de aplicaciones web.
- **Seguridad.** Debido a que las Aplicaciones Web están disponibles a través de acceso a la red, es casi imposible limitar la población de usuarios finales que pueden acceder a la misma. Con el fin de proteger el contenido sensible y proporcionar modos seguros de transmisión de datos, se deben implementar fuertes medidas de seguridad en toda la infraestructura que soporta una aplicación web, así como también en el nivel interno de la propia aplicación.

- **Estética.** En general, muchos sistemas web demandan una buena estética, poniendo énfasis en la creatividad visual y la incorporación de multimedia en la presentación e interfaces de usuario. Cuando una aplicación se diseña para comercializar o vender productos o prestar servicios que generan ingresos, la estética pueden tener tanto que ver con el éxito como el diseño técnico. Por lo tanto, la estética es un elemento clave para la aceptación del sistema.
- **Cambios tecnológicos.** Existen constantes avances en los estándares y tecnologías web. Es preciso en el desarrollo web hacer frente a nuevos lenguajes y herramientas, e incluso se debe manejar la inestabilidad de la tecnología en cuanto a errores y bugs que pueden aparecer en las primeras versiones de nuevos lenguajes, herramientas y entornos de desarrollo.
- **Diversas tecnologías.** El desarrollo web utiliza diversas tecnologías y estándares innovadores, e integra numerosos componentes variados, incluyendo software tradicional y no tradicional, lenguajes de scripting interpretados, archivos HTML, bases de datos, imágenes y otros componentes multimedia, como audio y vídeo, e interfaces de usuario complejas.
- **Medios de entrega.** El medio de entrega de contenido de aplicaciones web es bastante diferente del de software tradicional, ya que las mismas necesitan brindar respuesta frente a una variedad de dispositivos de visualización y formatos, y dar soporte a hardware, software y redes con velocidades de acceso muy diferentes.

2.2. INGENIERIA WEB

La Ingeniería Web es la aplicación de principios científicos, de ingeniería y de gestión y de enfoques disciplinados y sistemáticos para el desarrollo, implementación y mantenimiento exitoso de sistemas y aplicaciones

basadas en la web, de alta calidad (Murugesan, Deshpande, Hansen, & Ginige, 1999).

Se trata de un enfoque integral y proactivo para el desarrollo de aplicaciones web diversas y complejas, y su objetivo es mantener bajo control el potencial caos en el desarrollo de sistemas basados en la Web, minimizar los riesgos y mejorar la capacidad de mantenimiento y calidad de los sistemas Web.

Los primeros documentos sobre los problemas relacionados con la Ingeniería Web se publicaron en Conferencias de la World Wide Web entre los años 1995 y 1996. Luego, en el año 1998 en la 7^o Conferencia de la World Wide Web realizada en Brisbane, Australia; San Murugesan organizó el “Primer Workshop sobre Ingeniería Web”⁶. A partir de allí, la Ingeniería Web se está convirtiendo progresivamente en una nueva disciplina para abordar las necesidades y desafíos únicos del desarrollo de sistemas basados en la Web.

Teniendo en cuenta la naturaleza de la Web y de las aplicaciones basadas en la Web, la ingeniería Web se constituye como un campo multidisciplinario, abarcando aportes desde diversas áreas como ser: la interacción persona-computadora, interfaces de usuario, análisis y diseño de sistemas, ingeniería de software, ingeniería de requisitos, ingeniería hipermedia, estructuras de la información, pruebas, modelado y gestión de proyectos, como así también, se relaciona con las ciencias sociales, artes y diseño gráfico (Murugesan, Deshpande, Hansen, & Ginige, 1999). La Figura 2 ilustra la convergencia de las diferentes áreas sobre la Ingeniería Web:

⁶ Sitio Web de la Comunidad de Ingeniería Web: <http://www.webengineering.org>

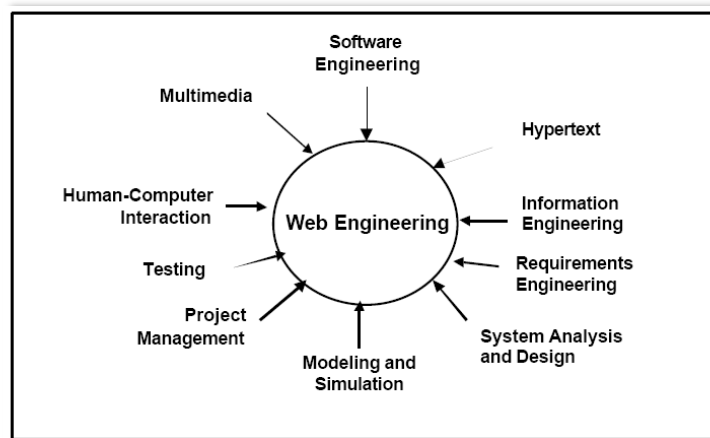


Figura 2: Ingeniería Web, un campo multidisciplinario

2.2.1. Actividades de la Ingeniería Web

La Ingeniería Web no es una sola actividad o tarea sino un proceso. Se ocupa de todos los aspectos del desarrollo de aplicaciones Web, a partir de la concepción y desarrollo de la implementación, evaluación del desempeño, y mantenimiento continuo. Por lo tanto, abarca las siguientes áreas de trabajo: (Murugesan, 2008)

- Obtención de requisitos y análisis;
- Modelado de sistemas Web;
- Arquitectura Web;
- Diseño de sistemas Web;
- Diseño de páginas web;
- Scripting / codificación;
- Interacción con bases de datos, sistemas ERP y otros sistemas basados en la Web;
- Calidad Web;
- Usabilidad Web;
- Seguridad Web;

- Evaluación del desempeño del sistema Web;
- Pruebas Web;
- Metodologías de desarrollo Web;
- Proceso de desarrollo Web;
- Métricas Web;
- Gestión de proyectos Web.

2.2.2. Características de Aplicaciones Web bien diseñadas

La Ingeniería Web se ha aplicado con éxito en una serie de aplicaciones web. Un sistema web bien diseñado tiene las siguientes propiedades: (Murugesan & Ginige, 2005)

- Funcionalmente completo y correcto
- Usable
- Robusto y confiable
- Mantenable
- Seguro
- Con rendimiento satisfactorio incluso con cargas repentinas y máximas
- Escalable
- Portable, cuando sea necesario ejecutar en diferentes plataformas comunes; compatible con múltiples navegadores
- Reutilizable
- Interoperable con otros sistemas Web y de información
- Accesibilidad universal (acceso de personas con diferentes tipos discapacidad)

- Bien documentado

2.3. MODELADO DE APLICACIONES WEB

Las disciplinas de ingeniería han utilizado con éxito modelos para reducir la complejidad, decisiones de diseño de documentos y facilitar la comunicación dentro de los equipos del proyecto.

El modelado tiene por objeto proporcionar una especificación del sistema que se construirá en un grado de detalle suficiente para su posterior implementación. El resultado del proceso de modelado son modelos que representan los aspectos relevantes del sistema en una manera simplificada e idealmente comprensible.

Durante el desarrollo de Aplicaciones Web, los modelos también constituyen un sólido punto de partida, teniendo en cuenta su complejidad y los aspectos estáticos y dinámicos de los niveles de contenido, hipertexto y presentación de una WebApp.

Con el paso de los años se han desarrollado diferentes enfoques de modelado para Aplicaciones Web que proporcionan la expresividad necesaria para modelar las características propias de las WebApp, a través de tres dimensiones: *niveles*, *aspectos* y *fases*, tal como se muestra en la Figura 3 (Kappel, Pröll, Reich, & Retschitzegger, 2003).

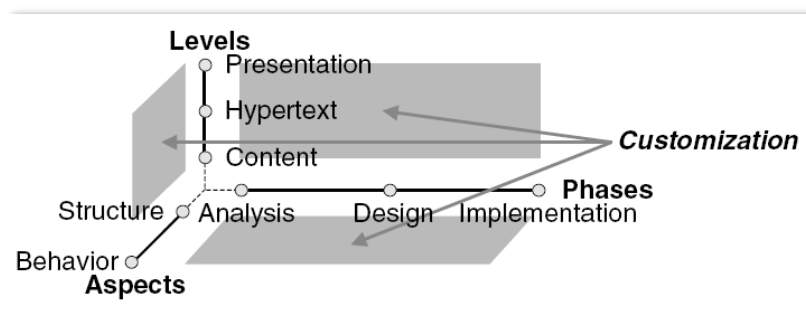


Figura 3: Requerimientos de modelado de Aplicaciones Web

Los modelos se construyen a lo largo de las diferentes “fases” de la ingeniería de requisitos, análisis, diseño e implementación correspondientes al proceso de desarrollo.

A su vez, estos modelos se utilizan para representar diferentes vistas de una misma aplicación web correspondiente a los distintos “niveles” que se detectan: contenido, estructura de navegación o hipertexto y presentación. El *modelo de contenido* se utiliza para especificar los conceptos que son relevantes al dominio de aplicación y la relación entre ellos. El *modelo de navegación* representa las rutas de navegación del sistema Web, se modela por separado del modelo de contenido, aunque deriva de éste. El *modelo de presentación* se enfoca en la creación de una estructura de presentación uniforme para las páginas que componen la Aplicación Web.

En la dimensión de “aspectos”, se modelan la *estructura* y el *comportamiento* para cada uno de los tres niveles de contenido, hipertexto y presentación, de acuerdo a los principios orientados a objeto. La relevancia de los modelos de estructura y comportamiento depende del tipo de Aplicación Web a ser implementada. Las aplicaciones web que principalmente manejan información estática requieren menos modelado del comportamiento en comparación con las aplicaciones web sumamente interactivas, como por ejemplo aplicaciones de comercio electrónico.

La dimensión de “adaptabilidad”, se refiere a un sistema web personalizado y dependiente del contexto que proporciona al usuario información más apropiada, vínculos o páginas dependientes de las características contextuales o del usuario. Esta dimensión influye sobre las demás dimensiones de modelado web.

2.3.1. Revisión de Métodos de Modelado

Se pueden destacar varios métodos de modelado que siguen diferentes paradigmas, algunos basados en cómo administrar sus datos, otros con el foco en el hipertexto, etc., tales como: WebML (Ceri, Daniel, & Matera, 2003), OOHDM (Schwabe & Rossi, 1998), UWE (Koch, 2001), OOWS (Pastor, Fons, Pelechano, & Abrahão, 2006), entre otros.

En la Figura 4 y Figura 5 se muestra el panorama de los métodos existentes en el ámbito de Ingeniería Web, indicando cuándo surgieron y en qué técnicas no específicas para la Web se basan (ER, OMT, UML), y cómo fueron evolucionando (Kappel, Pröll, Reich, & Retschitzegger, 2003):

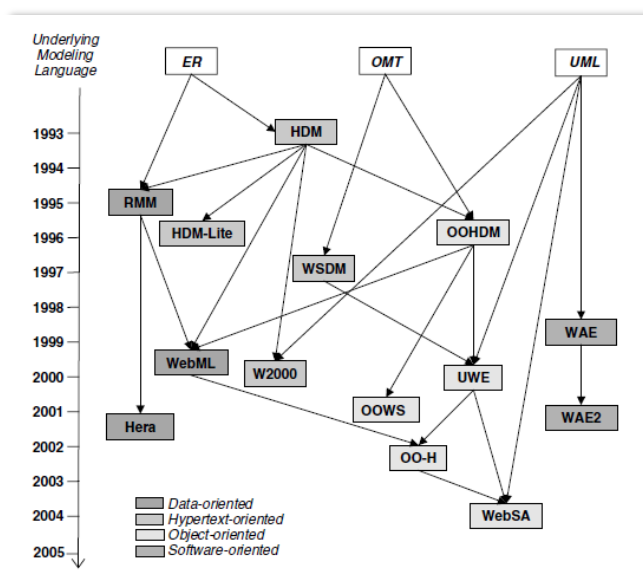


Figura 4: Desarrollo histórico de Métodos de Modelado de Aplicaciones Web

	Modeling Method	Modeling Paradigm	Notation	Evolving	Refinements Modeling	Content Modeling	Hypertext Modeling	Presentation Modeling	Customization Modeling	Structure and Behavior	Process / Approach	Tool Support	Generation	Strengths
HDM-lite	HT	ER + own notation	x	x	x	✓	✓	x	s	own	generation tools	auto		process for model transformation, automatic generation
Hera	DB	ER + RMM+ own notation	✓	x	✓	✓	✓	✓	s + b	own	authoring & generation tool	semi		model-driven development
OO-H	OO	UML + own notation	✓	✓	✓	✓	x	pers	s + b	own	modeling- & generation tool	auto		tool for automatic generation
OOHDM	OO	UML + own notation	✓	✓	✓	✓	✓	pers	s + b	own	x	x		powerful concepts for contextual navigation, personalization
OOWS	OO	UML + own notation	✓	✓	✓	✓	✓	x	s + b	own	modeling- & generation tool	auto		advanced (commercial) tool for automatic generation
RMM	DB	ER + own notation	x	x	✓	✓	✓	x	s	own	authoring tool	semi		hypertext modeling based on ER-model, predefined process
UWE	OO	UML	✓	✓	✓	✓	✓	pers	s + b	RUP	extended UML tool & generation tools	semi		UML-based method, model-driven development, aspect-oriented customization
W2000 (HDM)	HT	UML	✓	✓	x	✓	✓	pers	s	x	extended UML-tool	x		user-centric hypertext modeling
WAE2 (WAE)	SW	UML	✓	✓	✓	x	✓	x	s + b	RUP	standard UML-tools	x		implementation design, architectural design
WebML	DB	ER, UML	✓	✓	✓	✓	x	pers	s + b	own	modeling- & generation tool	auto		well-elaborated notation, database integration, generation
WS DM	HT	own notation	✓	x	✓	✓	x	x	s + b	own	x	x		user-centric approach for analysis

✓	supported	pers	personalization	RUP	Rational Unified Process	DB	data-oriented
x	not supported	s	structure modeling	own	own process model / approach	HT	hypertext-oriented
		b	behavior modeling	auto	automatic generation	OO	object-oriented
				semi	semi-automatic generation	SW	software-oriented

Figura 5: Revisión de Métodos de Modelado de Aplicaciones Web

De todos los métodos anteriores, se observa que no todos cubren los mismos aspectos. En algunos casos, hay métodos que no cubren todo el ciclo de vida, porque se especializan y tienen el foco en una fase, dejando sin gran soporte al resto de partes de la metodología. Por ejemplo, algunos métodos tienen el foco en el modelado, algunos en los requisitos y otros en la generación de código.

Sin embargo, tal como se aprecia en la Figura 5, UWE es uno de los métodos más completos, ya que cubre todos los aspectos de modelado,

siguiendo además el proceso de desarrollo “Proceso Unificado de Rational o R.U.P.” propuesto por Jacobson, Booch y Rumbaugh en 2001. Por este motivo, se ha elegido UWE para modelar la Aplicación Web propuesta en este trabajo.

2.3.2. Lenguaje de Modelado UML

UML⁷ (Unified Modeling Language) es un lenguaje de modelado estándar, respaldado por el Object Management Group⁸ (OMG) y es aplicable a la ingeniería de software permitiendo especificar, visualizar y documentar modelos de sistemas de software incluyendo su estructura, comportamiento y arquitectura, de manera que se cumplan todos sus requerimientos.

UML se utiliza normalmente como soporte a una metodología de desarrollo de software; por ejemplo RUP, sustentado en un grupo de componentes conceptuales como son actores, reglas, procesos de negocio, flujos de trabajo, componentes del sistema y actividades; y objetos concretos tales como diagramas, instrucciones de programación y componentes de software.

Los diagramas de UML brindan una perspectiva *estática* y *dinámica* del sistema; la primera hace énfasis en la estructura del sistema definiendo sus atributos, objetos, operaciones y relaciones; mientras que la segunda se enfoca más en la parte dinámica, es decir en las colaboraciones entre los objetos y en los cambios de estado de los mismos.

UML define un conjunto de diagramas, divididos dentro de tres categorías: diagramas de estructura, diagramas de comportamiento y diagramas de interacción.

⁷ Sitio Web de UML: <http://www.uml.org/>

⁸ Sitio Web de OMG: <http://www.omg.org/>

2.4. METODOLOGIA UML-BASED WEB ENGINEERING (UWE)

UWE es un enfoque de la ingeniería de software orientado al desarrollo de sistemas hipermedias⁹ en general y sistemas hipermedias adaptativos en particular (sistemas centrados en el usuario que se basan en el paradigma hipermedia, es decir, son una red de nodos conectados por enlaces, donde además se administra un modelo de usuario para adaptarse dinámicamente al mismo). Está basado en UML y el Proceso Unificado de Desarrollo de Software (RUP) (Koch, 2001).

Esta metodología surgió en el año de 1999 como un proyecto del Departamento de Programación e Ingeniería de Software de la Universidad Ludwig-Maximilians de Múnich, encabezado por Alexander Knapp, Gefei Zhang y Nora Koch, con la finalidad de estandarizar el proceso de análisis y diseño de sistemas Web, cubriendo totalmente su ciclo de vida y centrandolo su enfoque en aplicaciones personalizadas.

Este mismo grupo se encargó de la creación de las herramientas Open Source para el soporte de UWE como ArgoUWE¹⁰ y MagicUWE¹¹.

UWE está basado en estándares de la OMG como UML, Model Driven Architecture de OMG (MDA), Object Constraint Language (OCL) y eXtensible Markup Language (XML), asegurando su seguimiento mediante guías y especificaciones para el uso de tecnologías orientadas a objetos.

El principal objetivo del enfoque UWE¹² es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático;

⁹ Las aplicaciones web son consideradas un caso especial de sistemas hipermedias.

¹⁰ Sitio Web de ArgoUWE: <http://uwe.pst.ifi.lmu.de/toolargoUWE.html>

¹¹ Sitio Web de MagicUWE: <http://uwe.pst.ifi.lmu.de/toolMagicUWE.html>

¹² Sitio Web de UWE: <http://uwe.pst.ifi.lmu.de/index.html> - Institute of Programming and Software Engineering

y herramientas de soporte para la generación semi-automática de Aplicaciones Web.

La notación de UWE se define como una ligera extensión de UML, proporcionando un perfil UML para el dominio específico de la web.

2.4.1. Modelos de UWE

El método UWE consiste en la construcción de seis modelos de análisis y diseño (Koch, 2001). Dicha construcción se realiza dentro del marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos, diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación.

Los principales artefactos que produce el método de diseño de UWE son los siguientes:

- Un Modelo de Requerimientos que captura los requerimientos del sistema.
- Un Modelo Conceptual para el contenido (modelo de contenido).
- Un Modelo de Usuario.
- Un Modelo de Navegación que comprende la estructura de la navegación.
- Un Modelo de Presentación que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo del flujo de la presentación, modelo de interface abstracta de usuario, y modelo de ciclo de vida del objeto).
- Un Modelo de Adaptación.

La Figura 6 muestra los modelos de UWE representados como paquetes UML y sus relaciones de dependencias:

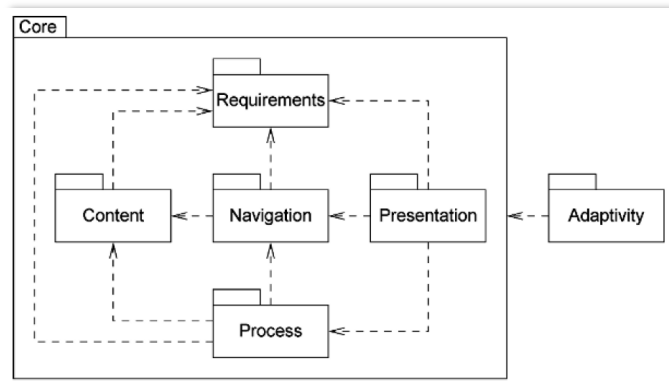


Figura 6: Meta-modelo de UWE

A continuación se hará una breve revisión de los modelos que propone UWE y que se tomarán en cuenta para este trabajo. Para más detalle remitirse a (Koch, Knapp, Zhang, & Baumeister, 2008) y (Koch, 2001).

2.4.1.1. Modelo de Requerimientos

El primer paso para el desarrollo de un sistema web que se especificará con UWE, es realizar la identificación de los requerimientos y plasmarlos en un *modelo de requerimientos*.

Los requerimientos pueden ser documentados en diferentes niveles de detalle, para este caso, UWE propone dos niveles de granularidad. En primera instancia se deben describir detalladamente las funcionalidades del sistema, las cuales son modeladas con casos de uso UML. Como segundo paso, se debe elaborar una descripción de los casos de uso más detallada, por ejemplo, realizando diagramas de actividad UML donde se delimiten las responsabilidades y acciones de los actores involucrados.

Los “casos de uso” fueron propuestos por el Proceso de Desarrollo de Software Unificado (RUP) para capturar los requerimientos del sistema. Es una técnica centrada en el usuario que obliga a definir quiénes son los usuarios (actores) de la aplicación y ofrece una forma intuitiva de

representar la funcionalidad que una aplicación tiene que cumplir para cada actor.

Los elementos de modelado principal utilizados en los casos de uso son: *actores* y casos de *uso*, donde estos últimos pueden tener una relación de *herencia*, *inclusión* o *extensión*. A su vez, por cada caso de uso se puede realizar una descripción textual detallada en términos de escenarios (primarios y secundarios).

Todos los elementos de modelado de casos de uso, como así también los mecanismos de vistas y paquetes son utilizados con la semántica definida en UML y representados gráficamente con la notación UML¹³.

Adicionalmente, UWE distingue tres tipos de casos de uso: *navegación*, *proceso*, y casos de uso *personalizados*.

Los casos de uso de *navegación*, se distinguen con el estereotipo <<navigation>> (□) y se utilizan para modelar el comportamiento típico del usuario cuando interactúa con una aplicación web, tal como navegar a través del contenido de la WebApp o buscar información por medio de palabras claves.

Los casos de uso de *proceso*, se utilizan para describir las tareas del negocio que los usuarios finales realizarán con el sistema, tales como acciones transaccionales sobre la Base de Datos. No se denota con ningún estereotipo específico, por lo tanto se utiliza en este caso la notación pura de UML.

Los casos de usos *personalizados*, implican la personalización de un sistema web, la cual es desencadenada por el comportamiento del usuario. Estos casos de uso se denotan con el estereotipo <<personalized>> (☆).

¹³ Versiones formales de UML: <http://www.omg.org/spec/UML/>

La Figura 7 muestra un ejemplo de diagrama de caso de uso para un caso de estudio de una Base de Datos de películas online, desde el punto de vista del usuario anónimo y usuario registrado:

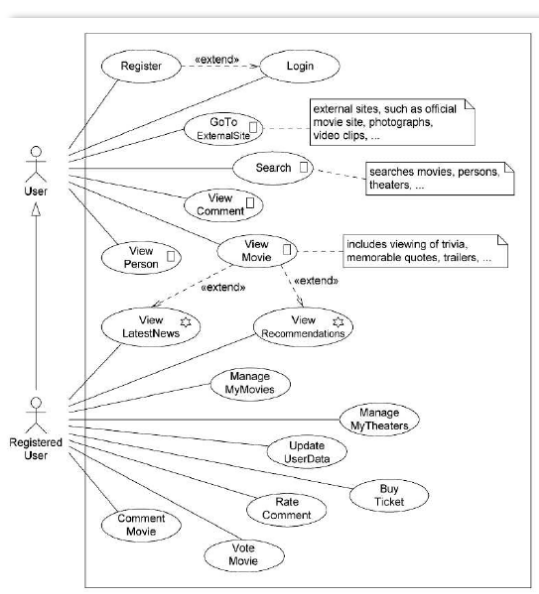


Figura 7: Modelo de Casos de Uso aplicando UWE

2.4.1.2. Modelo de Contenido (Conceptual) y Modelo de Usuario

El diseño conceptual se basa en el modelo de análisis e incluye los objetos involucrados en las actividades típicas que los usuarios realizan con la aplicación.

El propósito del *modelo de contenido* es proporcionar una especificación visual de la información relevante para el dominio del sistema web, que comprende principalmente el contenido de la aplicación Web.

También se puede construir un *modelo o perfil de usuario* con las entidades de dominio necesarias para Aplicaciones web personalizadas.

Los objetos que aparecen en la vista detallada de los casos de uso son los candidatos naturales de entidades del dominio para el modelo de contenido y de usuario.

Los principales elementos de modelado que se utilizan son: *clases* y *asociaciones*. Las clases definidas en este paso son utilizadas durante el diseño navegacional para derivar en nodos de la estructura hipermedia. Las asociaciones derivarán en links. Los modelos de contenido o de usuario son representados entonces como Modelos de Clases UML.

A continuación, en la Figura 8 y Figura 9 se muestra un ejemplo de modelo de contenido y otro de modelo de usuario para el caso de estudio de B.D. de películas online:

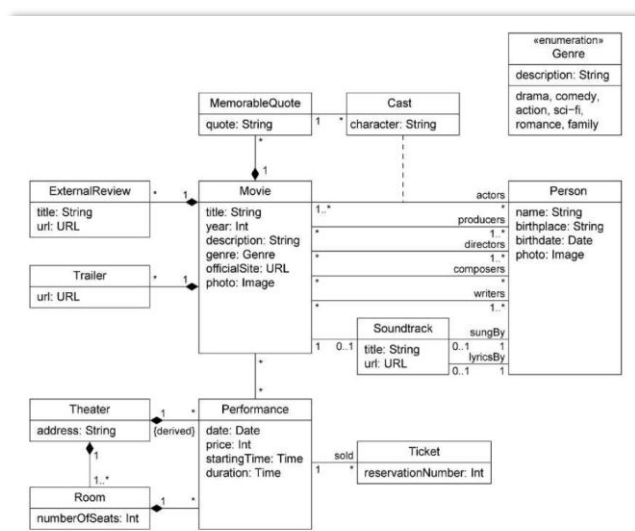


Figura 8: Modelo de Contenido para el caso de estudio de películas on-line

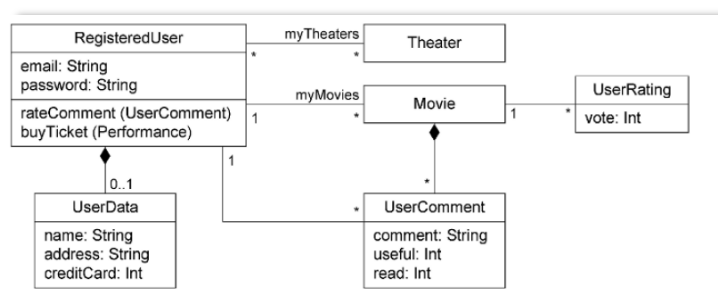


Figura 9: Modelo de Usuario para el caso de estudio de películas on-line

2.4.1.3. Modelo de Navegación

A partir del análisis de requerimientos y del modelado del contenido, se modela la *estructura de navegación* de la WebApp.

El modelo de estructura de navegación define la estructura de nodos y links de una WebApp mostrando cómo se puede realizar la navegación utilizando elementos de acceso tales como índices, visitas guiadas, consultas y menús.

Los elementos de modelado son:

- **Clases de navegación**, que se denotan con (□), representan los nodos navegables de la estructura de hipertexto.
- **Links de navegación**, que muestran el vínculo directo entre las clases de navegación.
- **Caminos de navegación alternativos**, los cuales son visualizados con el estereotipo <<menu>> (☰).
- **Primitivas de acceso**, las cuales se utilizan ya sea para llegar a múltiples instancias de una clase de navegación (<<index>> ☰ o <<guided tour>> ⇒) o para seleccionar ítems (<<query>> 🔍).
- **Clases de procesos** (Σ), las cuales modelan los puntos de entrada y de salida de los procesos de negocio. Cada clase de proceso está asociada a un caso de uso de proceso.
- **Links de procesos**, que representan el vínculo entre las clases de proceso y de navegación.

El modelo de *estructura de navegación* se representa mediante diagramas de clases UML estereotipados con las clases de navegación y procesos, menús y primitivas de acceso y así también los links de navegación y proceso.

En la Figura 10 se muestra un ejemplo del modelo de estructura de navegación para el caso de estudio mencionado.

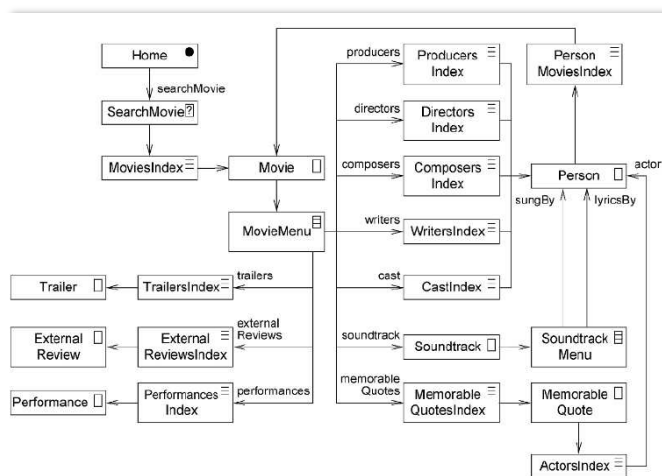


Figura 10: Fragmento de la estructura de navegación desde la Clase Películas

La estructura de navegación puede ser extendida con las clases de procesos que, como se definió anteriormente, representan los puntos de entrada y salida de los procesos de negocio. Estas clases de procesos derivan de los casos de uso no navegacionales.

A continuación, en la Figura 11 se muestra una parte del diagrama de navegación mostrado en la Figura 10, donde se integran los procesos de negocio *Register*, *Login* y *BuyTicket*.

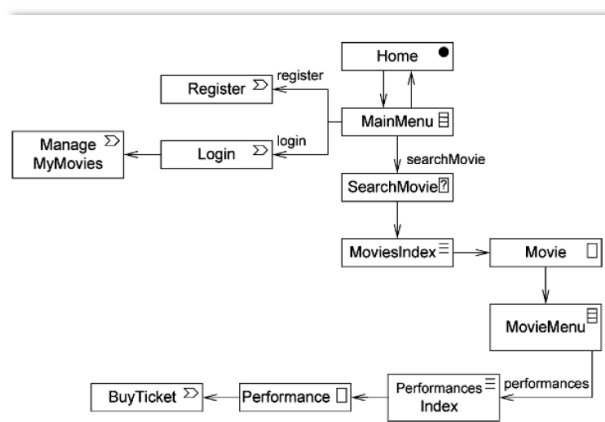


Figura 11: Integración de procesos de negocio a la navegación








Para evitar sobrecargas cognitivas en un mismo diagrama de navegación, se pueden producir diferentes vistas de la estructura de la navegación desde el modelo de contenido, haciendo énfasis en diferentes aspectos de la aplicación, tal como la navegación para un contenido en particular o la integración de los procesos de negocio relacionados.

2.4.1.4. Modelo de Presentación

El modelo de presentación proporciona una vista abstracta de la interfaz de usuario (UI) de la aplicación web. Se basa en el modelo de navegación y describe qué elementos (por ejemplo texto, elementos, links, formularios) se utilizarán para presentar los nodos de navegación.

La ventaja del modelo de presentación es que es independiente de las técnicas actuales utilizadas para implementar el sitio web, permitiendo que los desarrolladores debatan sobre los aspectos adecuados de la presentación antes de implementarlos.

Los elementos básicos del modelo de presentación son:

- **Clases de presentación**, las cuales se basan directamente en los nodos del modelo de navegación. Una clase de presentación () está compuesta por elementos de UI tales como, texto (`<<text>>` ) , vínculo (`<<anchor>>` ) , botón (`<<button>>` ) , imagen (`<<image>>` ) , formulario (`<<form>>` ) , y colección de vínculos (`<<anchored collection>>` )
- **Páginas web** (`<<page>>`), que se utilizan para modelar la información proveniente de varios nodos de navegación y que se presentan en una misma página web.
- **Grupo de presentación** (`<<presentation group>>`), el cual es un contenedor de clases de presentación, y a su vez de otros grupos de presentación.

En la Figura 12 se muestra un extracto del modelo de presentación de la página “películas”:

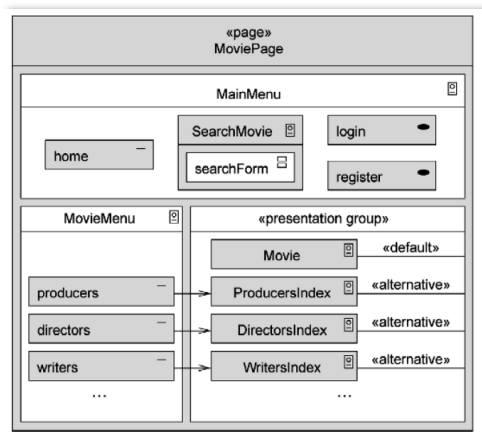


Figura 12: Modelo de presentación de la página Películas

2.4.1.5. Modelo de Adaptación

Un sistema hipermedia adaptativo, según (Brusilovsky, Schwarz, & Weber, 1996), es un sistema hipermedia que refleja algunas características del usuario en un *modelo de usuario* para luego proporcionar una adaptación dinámica de la hipermedia sobre la base del estado de dicho modelo de usuario. La adaptación consiste en cambios del contenido y/o de los nodos de presentación y enlaces. A su vez, un modelo de usuario se puede definir como una representación de características (conocimiento, preferencias, objetivos, etc.), que el sistema asume que un usuario posee.

Es importante diferenciar los *sistemas hipermedia adaptables* de los *sistemas hipermedia adaptativos*. Un sistema hipermedia adaptable es aquel que adecúa su comportamiento de acuerdo a cambios en la configuración de parámetros y preferencias que el usuario realiza explícitamente. En cambio, un sistema hipermedia adaptativo, como se mencionó en el párrafo anterior, se adapta de manera autónoma, monitoreando y registrando el comportamiento del usuario (su

navegación, sus respuestas a cuestionarios, etc.) en un modelo de usuario, y adaptando el sistema dinámicamente de acuerdo al estado actual del modelo de usuario (Bulterman, Rutledge, Hardman, & Van Ossenbruggen, 1999).

Los sistemas adaptativos en general (incluidos los hipermedia), pasan por diferentes etapas durante su utilización o se mantienen en diferentes estados con el fin interactuar con el usuario, adaptar la presentación y actualizar el modelo de usuario. Por el contrario, los sistemas de software no adaptativos únicamente oscilan entre la presentación y la interacción.

En la Figura 13 se muestra un modelo de ciclo de vida para la adaptación, representando los estados y posibles transiciones entre ellos, mediante un diagrama de estados UML.

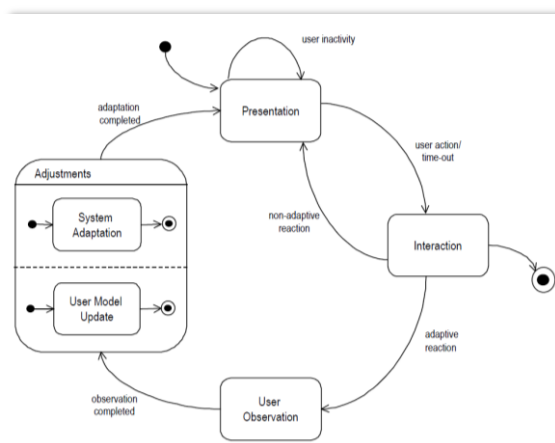


Figura 13: Modelo de ciclo de vida de la Adaptación

Los estados propuestos por los autores de UWE son: presentación, interacción, observación del usuario, y ajuste. A su vez, el estado “ajuste” se refina utilizando otros dos estados concurrentes: adaptación del sistema y actualización del modelo de usuario.

Como se aprecia en la Figura 13, además de las cuatro transiciones secuenciales: *acción del usuario*, *reacción adaptativa*, *observación* y *adaptación completada*, el modelo incluye las transiciones: *inactividad del*

usuario y reacción no adaptativa. El estado "inactividad del usuario" representa el sistema en espera de una acción del usuario, mientras que el estado "reacción no adaptativa", muestra que el sistema puede reaccionar también sin ajustar el modelo de usuario y la presentación.

UWE define en el "Modelo de Adaptación" las reglas de adaptación, especificando las condiciones bajo las cuales el contenido, navegación y presentación se adaptan; las acciones que se llevan a cabo en el proceso de adaptación y cómo se actualiza el modelo de usuario según las observaciones de su comportamiento. El modelo de adaptación se visualiza mediante diagramas de interacción UML (secuencia y colaboración) ya que éstos son apropiados para la representación dinámica del flujo de mensajes entre el modelo de adaptación y los otros modelos.

Los elementos de modelado que se utilizan para describir la funcionalidad adaptativa son:

- **Reglas**, que especifican cómo se encuentran los conceptos, cómo se construyen o presentan las páginas al usuario, así como la manera en que se actualiza el modelo de usuario. Se clasifican en: *reglas de construcción* (búsqueda de los conceptos apropiados); *reglas de adquisición* (para obtener información del usuario para actualizar el modelo de usuario); *reglas de adaptación* (para adaptar el contenido, enlaces y/o presentación). Una regla se modela mediante una condición, una acción y una función ejecutora y tiene el estereotipo <<rule>> con su correspondiente icono, tal como se muestra en la Figura 14:

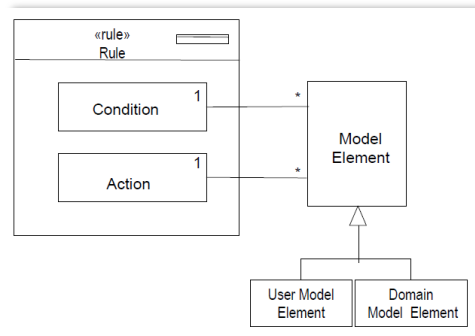


Figura 14: Modelado de la clase Regla

- **Comportamiento del usuario**, es una clase que modela el comportamiento del usuario según lo observado por el sistema y generalmente pueden ser de tres tipos en sistemas hipermedias: *navegación*; *ingreso de datos en formularios*; *inactividad del usuario* (registrada por un mecanismo de tiempo de espera). Esta clase es estereotipada con «user behaviour» y con su correspondiente icono. La Figura 15 muestra el patrón de análisis, mediante el cual se construye un modelo de adaptación a partir de reglas y comportamientos de los usuarios.

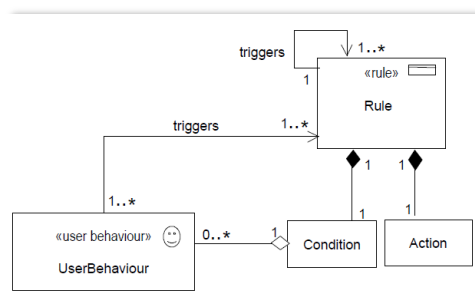


Figura 15: Patrón de adaptación

Cabe aclarar que si bien la especificación del enfoque UWE fue concebida inicialmente para modelar los aspectos de sistemas hipermedia adaptativos, sus autores afirman que también es apropiado para el desarrollo de aplicaciones web personalizadas y aplicaciones hipermedia no adaptativas, utilizando las técnicas y metodología presentada y

excluyendo las características del modelado de usuario y de la adaptación (Koch, 2001).

Al considerar que el prototipo de aplicación web propuesto no es un sistema adaptativo, de acuerdo a los requisitos funcionales y no funcionales planteados para el mismo, los modelos de usuario y adaptación no serán desarrollados en este trabajo final.

2.5. ACCESIBILIDAD WEB

La Accesibilidad Web significa que personas con algún tipo de discapacidad (visual, auditiva, física, cognitiva, neurológica y del habla) puedan percibir, entender, navegar e interactuar con la web, y a la vez que también puedan aportar contenidos a la web (Introduction to Web Accessibility, 2005).

Asimismo, también beneficia a otros grupos de usuarios como aquellas personas con dificultades relacionadas con el envejecimiento o las derivadas de una situación desfavorable determinada, como por ejemplo:

- Usuarios de edad avanzada con dificultades producidas por el envejecimiento.
- Usuarios afectados por circunstancias derivadas del entorno como baja iluminación, ambientes ruidosos, espacio reducido, etc.
- Usuarios con insuficiencia de medios que acceden a los servicios de Internet mediante equipos y conexiones con capacidades limitadas.
- Usuarios que no dominen el idioma, como aquellos de habla extranjera o con menor nivel cultural.
- Usuarios inexpertos o que presentan inseguridad frente a la utilización de diversos dispositivos electrónicos.

2.5.1. Pautas de la Iniciativa de Accesibilidad Web (WAI)

Con la misión de guiar la Web a su máximo potencial para ser accesible, permitiendo que las personas con discapacidad puedan participar en igualdad en la Web, la Iniciativa de Accesibilidad Web, perteneciente al Consorcio World Wide Web (W3C), desarrolla los estándares o pautas de accesibilidad web para diferentes componentes:

- **Pautas de Accesibilidad para Herramientas de Autor (ATAG).**
- **Pautas de Accesibilidad al Contenido Web (WCAG),** aborda el contenido Web, y es utilizado por desarrolladores, herramientas de autor y herramientas de evaluación de accesibilidad.
- **Pautas de Accesibilidad de Agentes de Usuario,** aborda los navegadores Web y reproductores multimedia, incluyendo algunos aspectos de las tecnologías de asistencia.

Es importante destacar que las pautas desarrolladas por WAI se basan en las especificaciones técnicas fundamentales de la Web tales como HTML, XML, CSS, etc.

En la Figura 16 se puede ver la interrelación entre los distintos componentes:

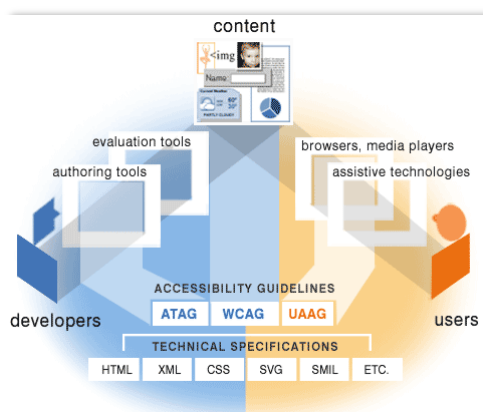


Figura 16: Relación entre los componentes y pautas de la Accesibilidad Web de WAI

En el desarrollo del presente trabajo, se tomarán en cuenta las Pautas de Accesibilidad al Contenido Web, por lo que a continuación se hará una breve reseña de las mismas.

2.5.2. Pautas de Accesibilidad al Contenido Web (WCAG)

Las Pautas de Accesibilidad para el Contenido Web, se desarrollan a través del proceso W3C con la cooperación de individuos y organizaciones de todo el mundo.

Los documentos de WCAG explican cómo hacer el contenido web más accesible para las personas con discapacidad, refiriéndose por “contenido web” a la información en una página o aplicación web, incluyendo información natural como texto, imágenes y sonidos; y el código o marcado que define la estructura, presentación, etc.

Las Pautas de Accesibilidad para el Contenido Web 2.0 (2008), es el último estándar técnico referenciable, fueron publicadas el 11 de diciembre de 2008 y sustituyen a la versión 1.0 de WCAG, que estaba disponible desde mayo de 1999. Las WCAG 2.0 se han diseñado para ser aplicadas a una amplia gama de tecnologías web ahora y en el futuro, y para ser verificables con una combinación de pruebas automatizadas y evaluación humana.

Para facilitar la aceptación de las normas por diversos Organismos Institucionales y Gubernamentales, las WCAG 2.0 se aprobaron como la Norma ISO/IEC 40500:2012 (WCAG 2.0), cuyo contenido es exactamente el mismo que el documento original de WCAG 2.0

Las WCAG 2.0 proporcionan varios niveles de orientación: principios, pautas, criterios de conformidad y técnicas suficientes y recomendables que actúan en conjunto para proporcionar una orientación sobre cómo crear un contenido más accesible (ver Figura 17).

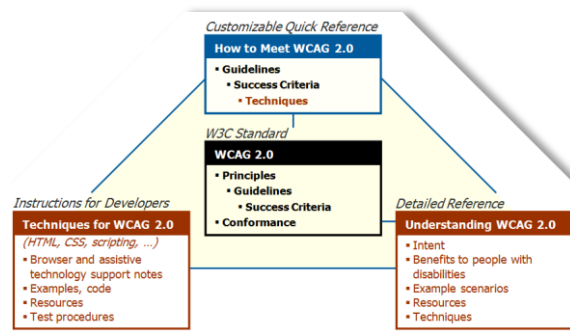


Figura 17: Documentos de WCAG 2.0

- **Principios:** en el nivel más alto se sitúan los cuatro principios que proporcionan los fundamentos de la accesibilidad web: *perceptible*, *operable*, *comprensible* y *robusto*.
- **Pautas:** por debajo de los principios están las 12 pautas que proporcionan los objetivos básicos que los autores deben lograr con el fin de crear un contenido más accesible para los usuarios con distintas discapacidades.
- **Criterios de Conformidad:** para cada pauta se proporcionan los criterios de conformidad verificables que permiten emplear las WCAG 2.0 en aquellas situaciones en las que existan requisitos y necesidad de evaluación de conformidad como: especificaciones de diseño, compras, regulación o acuerdos contractuales. Con el fin de cumplir con las necesidades de los diferentes grupos y situaciones, se definen tres niveles de conformidad: A (el más bajo), AA y AAA (el más alto).
- **Técnicas suficientes y recomendables:** para cada una de las pautas y criterios de conformidad, el grupo de trabajo ha documentado también una amplia variedad de técnicas. Las técnicas son informativas y se agrupan en dos categorías: aquellas que son suficientes para satisfacer los criterios de conformidad, y aquellas que son recomendables. Algunas de las técnicas recomendables tratan sobre barreras de accesibilidad que no han sido cubiertas por los

criterios de conformidad verificables. También se han documentado los errores frecuentes que son conocidos.

En cuanto a los principios, las WCAG 2.0 buscan alcanzar las siguientes metas:

- **Principio Perceptible:** La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que ellos puedan percibirlos.
- **Principio Operable:** Los componentes de la interfaz de usuario y la navegación deben ser operables.
- **Principio Comprensible:** La información y el manejo de la interfaz de usuario deben ser comprensibles.
- **Principio Robusto:** El contenido debe ser lo suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de aplicaciones de usuario, incluyendo las ayudas técnicas.

En la Figura 18 se muestra un mapa visual de las WCAG 2.0 donde se pueden apreciar los principios, pautas y criterios de conformidad para los niveles A, AA, AAA:

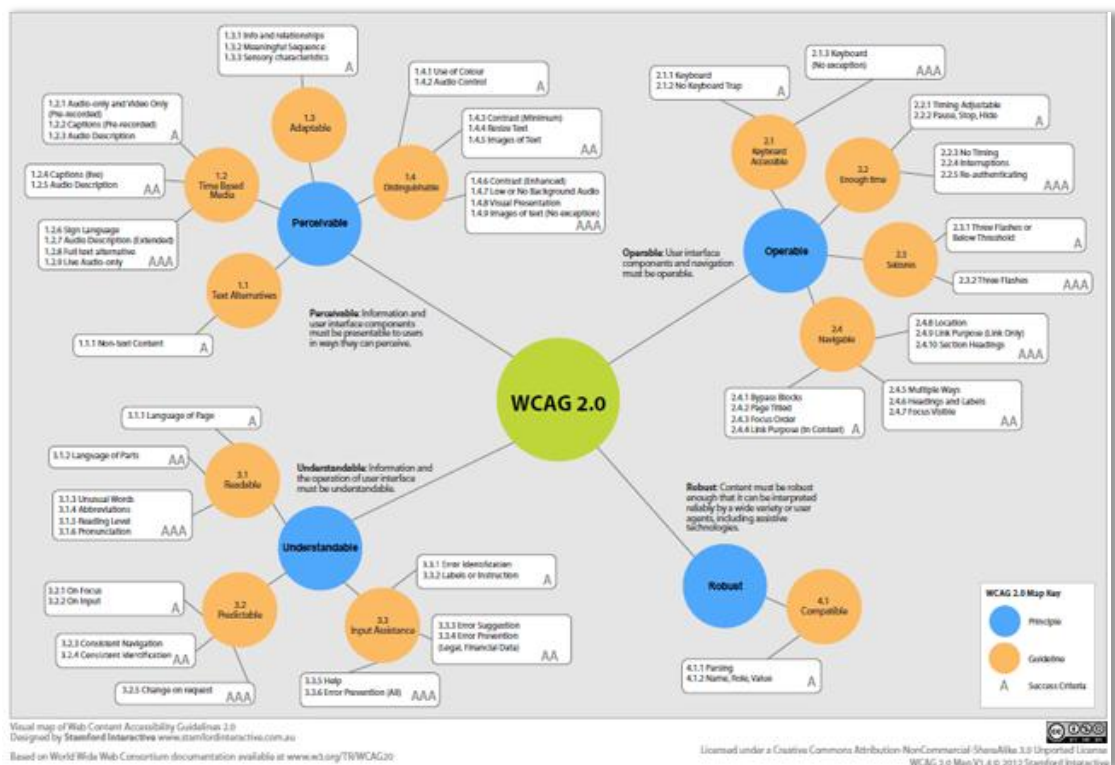


Figura 18: Mapa visual de las WCAG 2.0 - Fuente:

<https://www.digitalpulse.pwc.com.au/wp-content/uploads/2014/08/WCAG20Map.pdf>

En el Alcance del presente trabajo, se propuso realizar una “*identificación y aplicación de las directivas y técnicas de WCAG 2.0 para cumplir con el Principio Comprensible Nivel A de WCAG 2.0*”, por lo que a continuación se expondrán los aspectos relacionados a este principio.

2.5.1.1. Principio Comprensible de las WCAG 2.0

Este principio tiene por objetivo que la información y el manejo de la interfaz de usuario de las aplicaciones o sitios web sean comprensibles.

Posee 3 pautas y 5 criterios de conformidad nivel A, organizados de la siguiente manera:

Principio 3: Comprensible

PAUTA 3.1 Legible: Hacer que los contenidos textuales resulten legibles y comprensibles.

Criterios de Conformidad Nivel A:

3.1.1 **Idioma de la página:** El idioma predeterminado de cada página web puede ser determinado por software.

PAUTA 3.2 Predecible: Hacer que las páginas web aparezcan y operen de manera predecible.

Criterios de Conformidad Nivel A:

3.2.1 **Al recibir el foco:** Cuando cualquier componente recibe el foco, no inicia ningún cambio en el contexto.

3.2.2 **Al recibir entradas:** El cambio de estado en cualquier componente de la interfaz de usuario no provoca automáticamente un cambio en el contexto a menos que el usuario haya sido advertido de ese comportamiento antes de usar el componente.

PAUTA 3.3 Entrada de datos asistida: Ayudar a los usuarios a evitar y corregir los errores.

Criterios de Conformidad Nivel A:

3.3.1 **Identificación de errores:** Si se detecta automáticamente un error en la entrada de datos, el elemento erróneo es identificado y el error se describe al usuario mediante un texto.

3.3.2 **Etiquetas o instrucciones:** Se proporcionan etiquetas o instrucciones cuando el contenido requiere la introducción de datos por parte del usuario.

Para el diseño y codificación del prototipo de Aplicación Web propuesto se tomaron en cuenta las recomendaciones de técnicas suficientes de

WCAG 2.0 para cumplir con cada uno de los criterios de conformidad del nivel A.

2.6. GESTION DE LA CALIDAD EN LA WEB

2.6.1. Introducción a la Calidad de un Producto Software

La calidad de un producto software, es definida en el estándar ISO/IEC 9126 (1991) como: *“La totalidad de funciones y características de un producto software que permiten satisfacer las necesidades establecidas o implícitas”*.

De esta definición se puede apreciar que la calidad no es un concepto simple y atómico, sino un concepto abstracto, multidimensional y contextual que no puede ser medido y evaluado directamente. Por lo tanto, en la práctica la calidad se evalúa por medio de la cuantificación de conceptos de abstracción de más bajo nivel, tales como atributos de entidades. Los atributos pueden ser brevemente definidos como una propiedad medible de una entidad. Una entidad puede tener muchos atributos, pero solo algunos de ellos pueden resultar de interés para los propósitos de un proyecto de medición y evaluación dado.

Entonces calidad podría definirse como: *“una relación abstracta entre los atributos de una categoría de entidad y una necesidad de información específica, de acuerdo al objeto de la medición”* (Olsina, Covella, & Rossi, Web Quality, 2006).

La definición e implementación de estándares fijan los **atributos** deseables del software de calidad, a la vez que surgen modelos y metodologías para la evaluación de la calidad. Mediante estándares como ISO/IEC 9126-1 (2001) e ISO/IEC 25010 (2011) se ha tratado de determinar y categorizar los factores que afectan a la calidad del software

proponiendo un modelo de calidad de 3 vistas: Calidad Interna, Calidad Externa y Calidad en Uso para la evaluación de un producto de software.

La *Calidad Interna* puede ser medida y evaluada por atributos estáticos de un producto, como documentos de especificación de requisitos, porciones de código fuente, etc. La *Calidad Externa* se puede medir mediante propiedades dinámicas de un sistema, por ejemplo, módulo o aplicación completa que se ejecuta en un entorno cerca del real. Por último, la *Calidad en Uso*, es el grado en que un Software/Aplicación Web en uso, es utilizado por usuarios específicos ajustándose a sus necesidades para lograr objetivos de tarea específicos con eficacia, eficiencia, libre de riesgos y satisfacción en contextos específicos de uso. En la Figura 19 se puede ver la interrelación de los distintos niveles de calidad.

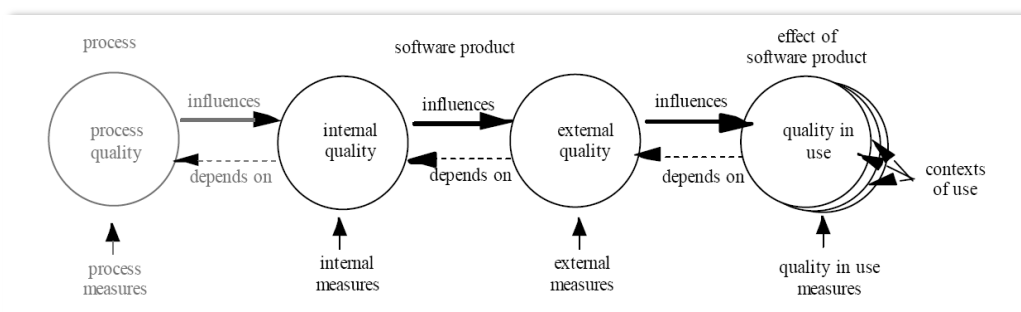


Figura 19: Calidad en el ciclo de vida (ISO/IEC 9126-1, 2001)

Por otra parte, el reciente estándar ISO/IEC 25010 (2011) define el conjunto de características y relaciones entre ellas que sirven de base para la especificación de requisitos de calidad y evaluación de la calidad, proponiendo 2 modelos de calidad: Calidad en Uso y Calidad de Producto Sistema/Software.

El modelo de *Calidad en Uso* está compuesto por 5 características y 9 sub-características que se relacionan con el resultado de la interacción cuando un producto se utiliza en un determinado contexto.

El modelo de *Calidad de Producto Sistema/Software*, el cual es objeto de estudio en este trabajo, abarca cualidades internas y externas del sistema y se compone de 8 características de calidad y 31 sub-características que se relacionan con propiedades estáticas de software y las propiedades dinámicas del sistema informático, tal como se muestra en la Figura 20. Las características y sub-características definidas por ambos modelos, proporcionan coherencia terminológica para especificar, medir y evaluar la calidad tanto de los productos software como de los sistemas informáticos.

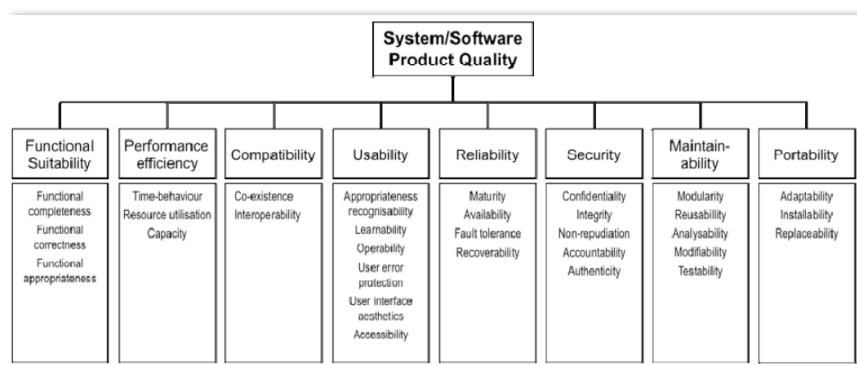


Figura 20: Modelo de Calidad de Producto sistema/software (ISO/IEC 25010, 2011)

En esta versión del estándar (ISO/IEC 25010, 2011), se ha reformulado la definición de la característica “Usabilidad” del modelo de Calidad de Producto Sistema/software, agregando, dos sub-características: *“Protección contra errores de usuario”* y *“Accesibilidad”*.

A su vez, la “Accesibilidad” es definida en este estándar como: *“grado en que un producto o sistema puede ser utilizado por personas con la más amplia gama de características y capacidades para lograr un objetivo determinado en un contexto de uso especificado”* (ISO/IEC 25010, 2011).

De esta manera, se concluye que la Accesibilidad no solo se considera desde un sentido práctico desde WAI, sino que además representa una sub-característica de la *Usabilidad* dentro del modelo de Calidad de Producto Sistema/software del estándar ISO.

2.6.2. Modelos de Calidad Web

Los sistemas basados en la Web, los cuales representan una combinación de información (contenido), funcionalidades y servicios integrados se convirtieron rápidamente en la forma más predominante de entrega de software en la actualidad, y por lo tanto requieren una atención más centrada en la comprensión de las diferentes características y requerimientos; evaluación; y sobre todo en la mejora de su calidad (Olsina, Lew, Dieser, & Rivera, 2012).

Como se vio en secciones anteriores, existen características propias de las Aplicaciones Web que se diferencian del software tradicional, y a su vez, existen diferentes categorizaciones de sistemas basados en la Web, lo que convierte a cada Aplicación Web en un artefacto particular.

Sin embargo, al igual que aplicaciones software tradicionales, los sistemas web también involucran código fuente y ejecutable, bases de datos, y especificaciones de requerimientos, de arquitectura, de diseño y pruebas.

Por consiguiente, algunos trabajos como en (Olsina, Covella, & Rossi, Web Quality, 2006), sostienen que los modelos de calidad ISO son también aplicables en gran medida a los productos resultantes del ciclo de vida del desarrollo web. Aunque, en algunos casos, es necesario adaptar los modelos de calidad de acuerdo a las particularidades específicas de los requerimientos de calidad web.

Siguiendo este camino, en (Olsina, Lew, Dieser, & Rivera, 2012) se propone la versión 2.0 de un framework de modelado de calidad web denominado 2Q2U (Quality, Quality in Use, actual Usability and User experience), el cual es una extensión del framework y modelos de calidad de ISO 25010 y además, es una actualización de la versión 1.0 de 2Q2U, donde se agregan características y sub-características de acuerdo a la nueva generación de Aplicaciones Web (ver Figura 21).

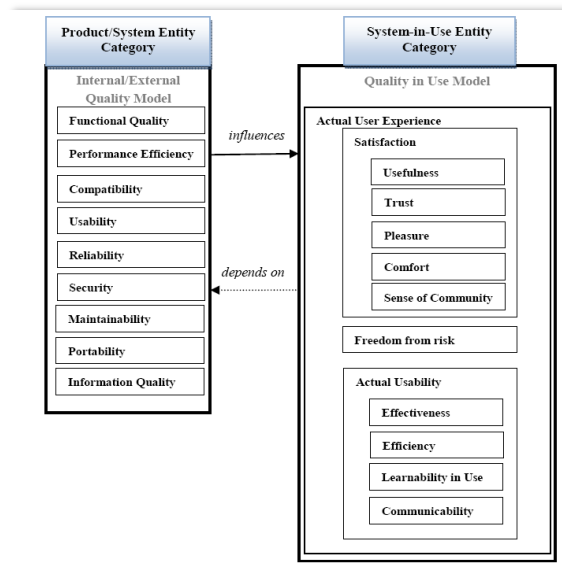


Figura 21: Características del modelo de calidad y relación entre las vistas del framework 2Q2U v2.0

✦ Tanto en el modelo de ISO/IEC 25010 (2011) como en el modelo propuesto por 2Q2U v2.0, se define a la *Accesibilidad* como una sub-característica de *Usabilidad*, no especificando ninguna sub-característica ni atributos al nivel de *Accesibilidad*. Por lo tanto, en este trabajo se propone un modelo de sub-características y atributos medibles para evaluar el concepto calculable (abstracto) *Accesibilidad*, el cual será construido de acuerdo a las Pautas de WCAG 2.0.

2.6.3. Estrategia de Medición y Evaluación (M&E) de Calidad GOCAME

Para realizar la evaluación de la calidad de un producto software, no basta solamente con la definición de los modelos de calidad revisados en las secciones anteriores. Estos modelos deben ser instanciados, es decir, se deben considerar las características, atributos y relaciones que sean relevantes para un propósito dado y una necesidad de información de una categoría de entidad en concreto. Entonces, para impulsar un proceso de

medición y evaluación de la calidad acorde a los requerimientos, se debe adoptar una estrategia específica.

Para medir y evaluar la característica “Accesibilidad” del prototipo de Aplicación Web propuesto en este trabajo, se utilizará la estrategia integrada de Medición y Evaluación GOCAME (Goal-Oriented Context-Aware Measurement and Evaluation) (Olsina, Papa, & Molina, 2008).

GOCAME es una estrategia que posee un enfoque orientado a objetivos en el cual todas las actividades son guiadas por un estado y necesidad de información específica (Olsina, Lew, Dieser, & Rivera, 2012). Dicha necesidad de información intenta satisfacer requerimientos no funcionales de alguna categoría de entidad para un propósito particular y punto de vista de las personas interesadas. Los requerimientos no funcionales son representados por modelos de conceptos que incluyen conceptos calculables de alto nivel como por ejemplo los modelos de calidad del framework 2Q2U mencionado anteriormente. Los modelos de calidad instanciados representan la fuente de datos para las actividades de medición y evaluación. La *medición* se especifica e implementa utilizando métricas, las cuales definen cómo representar y recolectar los valores de atributos. La *evaluación* se especifica e implementa utilizando indicadores, los cuales definen cómo interpretar los valores de los atributos y a la vez permiten calcular los conceptos calculables de alto nivel del modelo de calidad. Finalmente los datos e información proveniente de las actividades de medición y evaluación se utilizan para las etapas de análisis y recomendación.

La estrategia GOCAME soporta de manera simultánea tres capacidades (Becker, Papa, & Olsina, 2013):

- I. Un *marco conceptual* centrado en una base terminológica y estructurado en seis componentes denominado **C-INCAMI** (Contextual-Information Need, Concept model, Attribute, Metric and Indicator)

(Olsina, Papa, & Molina, 2008) que permite especificar los datos y metadatos utilizados en las actividades y artefactos del proceso, como así también, instanciar métodos y herramientas de soporte para automatizar todo o parte del proceso (ver Figura 22). Los componentes de C-INCAMI son: *Definición de proyecto de M&E, Especificación de requerimientos No Funcionales, Especificación del Contexto, Diseño de la Medición e implementación, Diseño de la Evaluación e implementación; y Especificación de Análisis y Recomendación.*

- II. *Especificaciones de proceso* de M&E desde diferentes puntos de vista, es decir, perspectivas funcionales, organizacionales, de comportamiento y de información (Becker, Lew, & Olsina, 2012). La especificación de proceso de M&E de GOCAME, se compone de seis actividades principales, como se muestra en la Figura 23. Estas actividades son las siguientes: (A1) Definir los requisitos no funcionales, (A2) Diseñar la medición, (A3) Implementar la medición; (A4) Diseñar la evaluación; (A5) Implementar la Evaluación, y (A6) Analizar y recomendar.
- III. La *metodología WebQEM* (Web Quality Evaluation), que proporciona el "cómo" definir los requerimientos no funcionales, la medición, evaluación, análisis y actividades de recomendación. Se compone de un conjunto de métodos, técnicas y herramientas para llevar a cabo la descripción de las actividades.

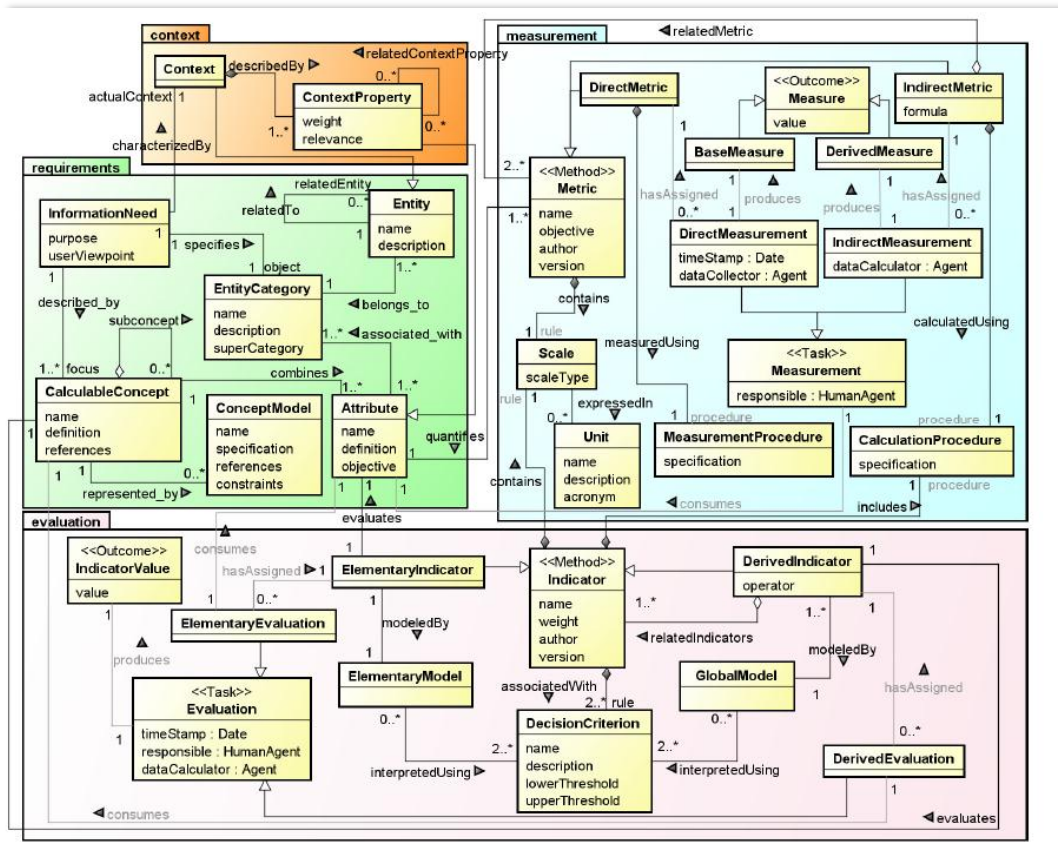


Figura 22: Principales conceptos y relaciones de los componentes del marco C-INCAMI

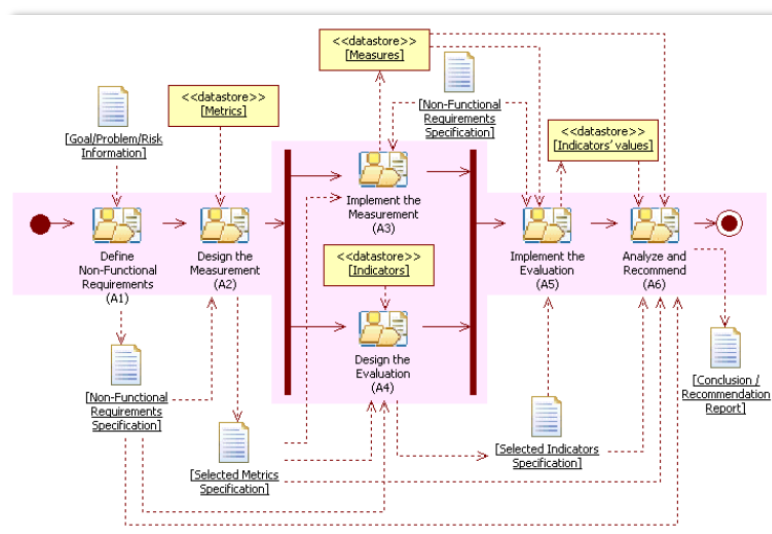


Figura 23: Puntos de vista funcional y de comportamiento del proceso de GOCAME

2.7. TECNOLOGIAS DE DESARROLLO

En esta sección se describirán brevemente los conceptos y tecnologías que se utilizaron para el desarrollo y codificación del prototipo de Aplicación Web propuesto en este trabajo.

En la Tabla 1 se muestra un resumen de las tecnologías software de acuerdo a su finalidad en el desarrollo de prototipo de Aplicación Web.

Tabla 1: Tecnologías software de desarrollo Web

Nombre Tecnología	Finalidad
NetBeans IDE	Entorno de desarrollo de aplicación
PostgreSQL	Sistema Gestor de Base de Datos
Groovy	Lenguaje de Programación
Grails	Framework de Desarrollo Web

2.7.1. Entorno de Desarrollo Integrado (IDE) Netbeans

NetBeans IDE¹⁴ es un *entorno de desarrollo integrado*, que ofrece una interfaz gráfica basada en ventanas y asistentes para la creación de componentes, permitiendo a los desarrolladores de software escribir, compilar, depurar y ejecutar aplicaciones. Está escrito en Java, pero brinda la posibilidad de trabajar con otros lenguajes de programación para la creación de aplicaciones de escritorio, web, móviles, etc. Al funcionar mediante la tecnología Java, NetBeans es portable, es decir que no se limita a un sistema operativo en particular.

NetBeans IDE es un producto de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2.

¹⁴ Sitio Web de Netbeans: <https://netbeans.org/>

2.7.2. Gestor de Base de Datos PostgreSQL

PostgreSql¹⁵ es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y exactitud. Se ejecuta en la mayoría de los sistemas operativos, tales como Linux, UNIX, Windows.

Algunas características destacadas de PostgreSQL son:

- ✓ Totalmente compatible con la prueba ACID (Atomicidad, Consistencia, Integridad, Durabilidad)
- ✓ Tiene soporte completo para: claves foráneas, relaciones, vistas, triggers (disparadores) y procedimientos almacenados (en múltiples lenguajes)
- ✓ Soporta la mayoría de tipos de datos del estándar SQL:2008.
- ✓ También es compatible con el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o vídeo.
- ✓ Cuenta con interfaces de programación nativas para C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otros.
- ✓ Posee características sofisticadas como Control de Concurrencia Multi-Versión (MVCC), es altamente escalable tanto en la enorme cantidad de datos que puede manejar y en el número de usuarios simultáneos que puede atender.

2.7.3. Lenguaje de Programación Groovy

A través de los años la plataforma Java se fue expandiendo hasta cubrir un amplio espectro, desde aplicaciones empresariales hasta aplicaciones móviles y embebidas. La fortaleza de Java reside por una parte, en el rico conjunto de APIs (Application Programming Interfaces) los cuales

¹⁵ Sitio Web de PostgreSQL: <http://www.postgresql.org/>

representan librerías de componentes de código abierto que resuelven problemas de diferentes dominios y que están disponibles para ser utilizados desde cualquier proyecto de software. Otra potente característica es su Máquina Virtual (Java Virtual Machine - JVM), la cual se considera una interfaz estandarizada que permite ejecutar una aplicación escrita en Java, sobre cualquier arquitectura de hardware y sistema operativo. Cabe destacar que en los años recientes, se han introducido nuevos lenguajes en los que puede ser codificada una aplicación y ejecutada por la JVM, tales como *JRuby* (una implementación del lenguaje Ruby) y *Jython* (una implementación del lenguaje Python y Groovy).

El lenguaje “Groovy”, es también un lenguaje basado en Java, pero difiere de los lenguajes mencionados anteriormente en que éste fue diseñado específicamente para ser código-compatible con el lenguaje Java, siendo además, binario-compatible a nivel de código byte (Dearle, 2010). El proyecto Groovy se originó en Agosto de 2003, inspirado en otros lenguajes dinámicos existentes como Ruby, Python, Smalltalk; con el objetivo de proporcionar un lenguaje dinámico y orientado a objeto, el cual puede ejecutarse sobre la JVM.

2.7.3.1. Características del Lenguaje Groovy

Groovy agrega características únicas que lo distinguen del lenguaje Java y permite a los desarrolladores crear código de alto nivel, y utilizar un idioma más abstracto que en Java. A continuación se mostrará un breve resumen de las características más relevantes de Groovy según Dearle (2010) son:

- **Tipificación de datos estática y opcional:** en Groovy, el tipo de dato de una variable puede ser *determinado* en tiempo de asignación, al contrario de Java donde el tipo de dato debe ser *declarado* antes de

asignar un valor a una variable (tipeo estático). Groovy soporta tanto el tipado de datos estático como opcional (dinámico).

- **Soporte nativo para listas y mapas:** Groovy agrega soporte nativo para todos los tipos de colecciones de Java, mediante una sintaxis más intuitiva y legible, a diferencia de Java que requiere del uso de interfaces para la manipulación de listas y mapas.
- **Closures:** son fragmentos de código anónimos que pueden ser asignados a una variable para luego ser ejecutado. Los closures pueden contener múltiples sentencias y puede ser tan complejo como uno desee.
- **Sobrecarga de operador Groovy:** cualquier clase Groovy puede contener un conjunto completo de operadores implementando el método correspondiente apropiado en la clase. Por ejemplo, el operador + (más) es implementado mediante el método “*plus()*”.
- **Soporte para expresiones regulares:** Groovy construye expresiones regulares internamente dentro del lenguaje mediante el operador “*=~*” y objetos “*matcher*”.
- **Marcado Groovy:** Groovy posee clases constructoras que generan directamente scripts de HTML, XML, entre otros, con una estructura basada en árbol, simplemente escribiendo código Groovy.

En resumen, Groovy toma los beneficios del lenguaje Java y agrega características de productividad y flexibilidad ofrecidas por los lenguajes dinámicos. Por otra parte, Groovy también ha demostrado ser una gran plataforma de soporte para áreas como meta-programación y lenguajes específico de dominio.

2.7.4. Frameworks de Desarrollo Web

En el desarrollo de software, un framework o marco de trabajo, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

En otras palabras, el término framework, se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Es decir, que un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones de diseño.

Un “framework Web”, por tanto, se puede definir como un conjunto de componentes que constituyen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. Por lo general, la mayoría de los frameworks Web se basan en el patrón de diseño arquitectónico “Modelo Vista Controlador”, del cual se hará una breve reseña en la siguiente sección.

2.7.5. Patrón Modelo-Vista-Controlador

El Modelo–Vista–Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son: *el modelo, la vista y el controlador*, es decir, por un lado define componentes para la representación de la

información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento (Modelo–vista–controlador).

Por lo general, los componentes de MVC de una Aplicación Web, se podrían definir como sigue¹⁶:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio (las base de datos pertenecen a esta capa).
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

En la Figura 24 se puede apreciar una representación del patrón MVC.

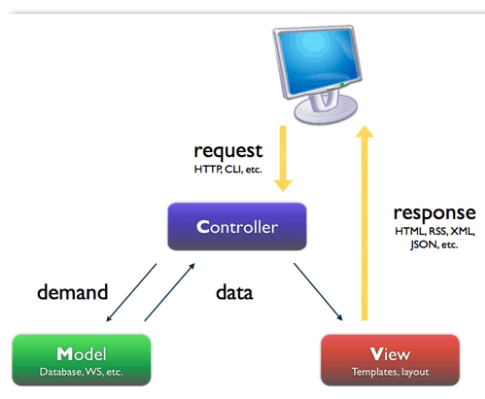


Figura 24: Arquitectura MVC (Fuente:

http://librosweb.es/libro/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html)

2.7.6. Framework de Desarrollo Grails

La construcción de aplicaciones web modernas es una tarea compleja ya que involucra en su entorno diversas tecnologías tales como Lenguaje de

¹⁶ MVC en: http://librosweb.es/libro/symfony_1_2/capitulo_2/el_patron_mvc.html

Marcado de Hipertexto (HTML), Hojas de Estilos en Cascada (CSS), JavaScript Asíncrono y XML (Ajax), XML, servicios web, bases de datos, etc. A su vez, por sobre las tecnologías, existen una gran variedad de opciones de framework de código abierto tales como frameworks modelo-vista-controlador (MVC) y frameworks Ajax.

En años recientes la comunidad Java ha tratado de resolver las cuestiones relacionadas al desarrollo de aplicaciones web, utilizando la plataforma Java Edición Empresarial (Java EE) y su predecesor, Java 2 Edición Empresarial (J2EE). Si bien estas plataformas son escalables y robustas, no permiten un desarrollo rápido y ágil. Por esta razón, se crearon frameworks que ganaron popularidad a través del tiempo tales como Struts, Spring, e Hibernate. Aun así, el ciclo de desarrollo de codificación, compilación, empaquetado, despliegue, testing, y depuración, toma demasiado tiempo para cualquier productividad real y requiere que los desarrolladores cambien de contexto frecuentemente.

En este punto se introduce “Grails”¹⁷, el cual es un framework de desarrollo web para la plataforma Java, que aumenta considerablemente la productividad de los desarrolladores a través de la confluencia de un lenguaje dinámico, una filosofía de convención sobre configuración, herramientas de apoyo poderosamente pragmáticas, y una perspectiva ágil procedente de los mejores paradigmas de desarrollo web emergentes (Smith & Ledbrook, 2009).

2.7.6.1. Características destacadas del Framework Grails

Las características más importantes de Grails son (Judd, Faisal Nusairat, & Shingler, 2008):

¹⁷ Sitio Web de Grails: <https://grails.org/>

- **Patrón Modelo-Vista-Controlador:** Grails es un framework que se basa en el patrón arquitectónico Modelo-Vista-Controlador (MVC), donde interactúan los siguientes componentes:
 - ✓ *Clases de Dominio*, las cuales representan el *Modelo* y contienen los datos que se muestran en la vista (interfaz de usuario), y que a la vez son automáticamente persistentes en una base de datos.
 - ✓ *Controladores*, que manejan las peticiones del usuario y orquestan servicios y otros comportamientos.
 - ✓ *Groovy Server Pages (GSP)*, la cual es una tecnología que genera las *Vistas* de la aplicación web generalmente en el lenguaje HTML, proporcionando una estructura flexible y librerías de etiquetas (en inglés: tags).
- **Convención sobre Configuración:** Grails minimiza la utilización de archivos de configuración XML, en lugar de eso se basa en convenciones de nombres de archivos, estructura de directorios, etc., contribuyendo de esta manera al cumplimiento del principio “*No te repitas (Don’t Repeat Yourself - DRY)*”
- **Testing Unitario:** ayuda a mejorar la calidad del software y permite la mantenibilidad a largo plazo de una aplicación. A su vez, en lenguajes de tipado dinámico como Groovy, los testing unitarios permiten identificar los efectos de cambios que se hagan en el código de la aplicación. En Grails, los test unitarios son creados automáticamente cuando se generan las clases de dominio o las clases controladores.
- **Scaffolding:** esta característica permite a Grails, con unas pocas líneas de código, generar aplicaciones con la funcionalidad de “*crear, leer, actualizar y eliminar (CRUD)*”, permitiendo al desarrollador focalizarse en la definición de propiedades, comportamiento y restricciones de las clases de dominio. Ya sea en tiempo de ejecución o en tiempo de desarrollo, Grails puede generar el comportamiento del

controlador y las vistas GSP asociadas con las clases de dominio para la funcionalidad CRUD. Al mismo tiempo, también puede generar un esquema de base de datos incluyendo las tablas para cada clase de dominio.

- **Mapeo Objeto-Relacional:** Grails incluye un potente framework de mapeo objeto relacional denominado “Grails Object Relational Mapping (GORM)”, el cual permite mapear los objetos a la base de datos relacional y representar las relaciones entre ellos. A diferencia de otros frameworks de mapeo objeto relacional, GORM inyecta los métodos CRUD correspondientes dentro de las clases sin tener que implementarlos. También proporciona métodos y criterios de búsqueda dinámicos.
- **Plug-Ins:** Grails proporciona una arquitectura de plug-ins para añadir fácilmente alguna funcionalidad compleja a nuestra aplicación, tal como seguridad, Ajax, testing, búsquedas, etc.
- **Tecnologías de código abierto integradas:** Grails integra lo mejor de la industria de estándares tales como: el lenguaje dinámico *Groovy*; el *Framework Spring*¹⁸ que proporciona un nivel de abstracción de aplicación superior al del API Java EE; *Hibernate*¹⁹ el cual es un potente framework de persistencia objeto-relacional que proporciona los fundamentos de GORM; *SiteMesh*²⁰ el cual es un framework de diseño de páginas web que implementa el patrón de diseño “decorator” para el renderizado HTML; el framework *JUnit*²¹ para la implementación del testing unitario; entre otros.

¹⁸ Sitio Web de Spring: <http://spring.io/>

¹⁹ Sitio Web de Hibernate: <http://www.hibernate.org>

²⁰ Sitio Web de SiteMesh: <http://wiki.sitemesh.org/wiki/display/sitemesh/Home>

²¹ Sitio Web de JUnit: <http://www.junit.org>

A continuación, en la Figura 25 se ilustra la pila (*stack en inglés*) de tecnologías y componentes que integran Grails (Rocher & Brown, 2009):

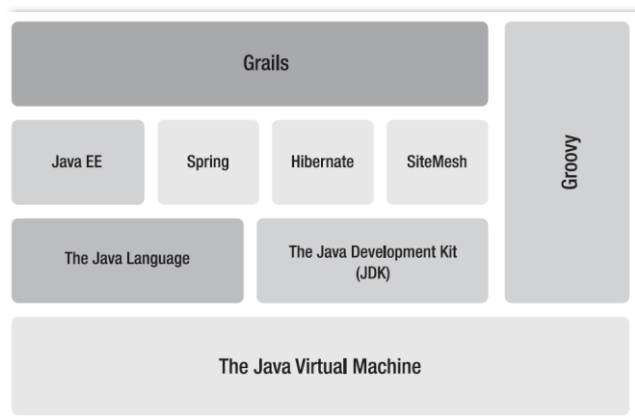


Figura 25: Stack de componentes en Grails

CAPITULO III - DISEÑO DEL MODELO DE ATRIBUTOS DE CALIDAD

3.1. INTRODUCCION

Con el objetivo de realizar el análisis y modelado del Prototipo de Aplicación Web considerando desde el inicio los atributos de accesibilidad web que la misma debe satisfacer, se decidió alterar el orden de las actividades planteadas en un primer momento para este trabajo.

Por lo tanto, en el presente capítulo se abordarán las actividades de la Estrategia de Medición y Evaluación de Calidad GOCAME que se refieran al diseño de la medición de la calidad del Aplicativo Web, es decir, el modelo de atributos y métricas e indicadores asociados. Para ello se ejecutarán las siguientes actividades planteadas en el proceso de GOCAME: (A1) Definir los requisitos no funcionales, (A2) Diseñar la medición y (A4) Diseñar la evaluación.

En el siguiente capítulo, se expone el informe de requisitos y modelado del prototipo web, donde se han tomado en cuenta las características y atributos de accesibilidad al contenido web definidas en este capítulo y que luego de finalizado el desarrollo de la WebApp, serán evaluadas implementando las restantes actividades de GOCAME: (A3) Implementar la medición, (A5) Implementar la Evaluación, y (A6) Analizar y recomendar.

En resumen, la Figura 26 muestra un diagrama de las actividades generales involucradas en el desarrollo del presente Trabajo Final, donde se puede apreciar que durante el modelado y desarrollo de la WebApp, se consideraron las especificaciones del modelo de atributos y métricas resultantes de etapas anteriores. Cabe aclarar que el proceso de desarrollo de la webapp se realizó de manera iterativa e incremental,

retornando a etapas anteriores en los casos que fuera necesario, incluso redefiniendo métricas o atributos del modelo de calidad, o bien, ajustando los modelos y/o diseño de las páginas web para que satisfagan los requerimientos de accesibilidad al contenido web.

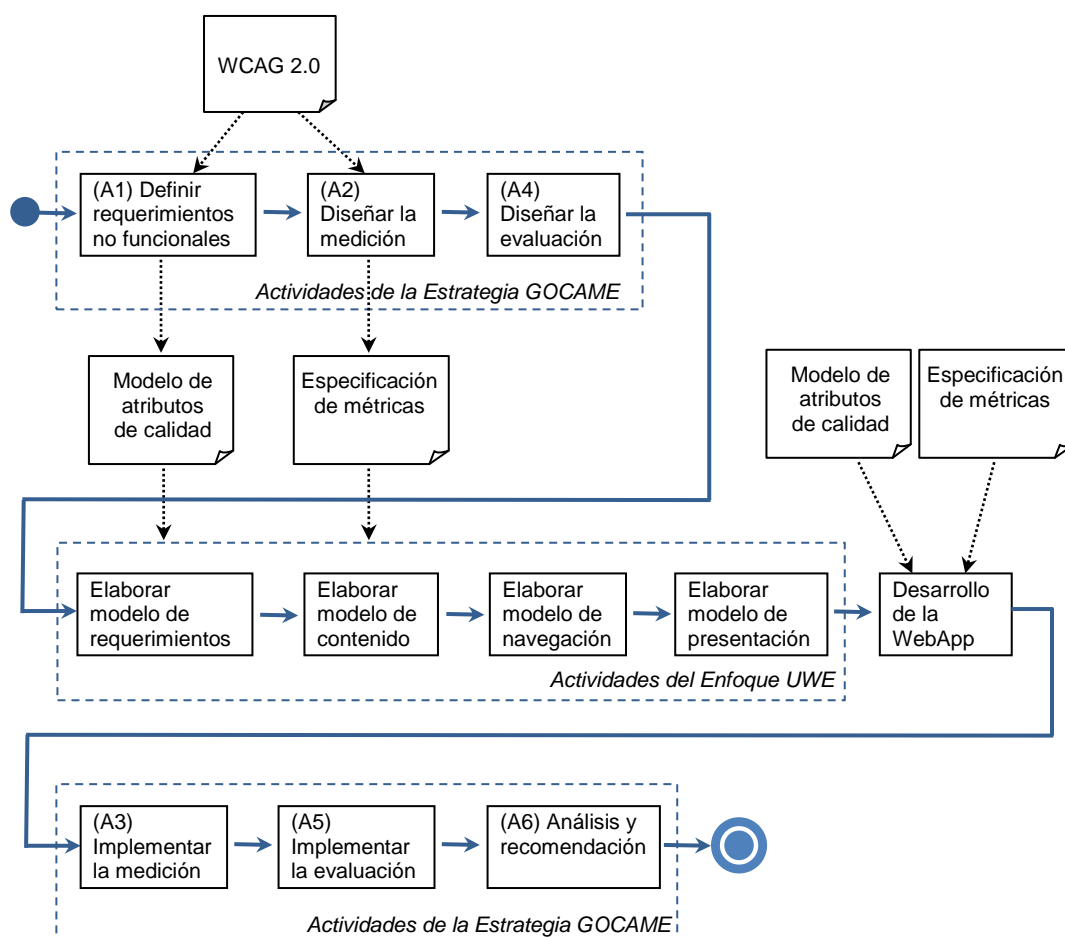


Figura 26: Secuencia de Actividades desarrolladas en el presente Trabajo Final

3.2. (A1) DEFINICION DE REQUERIMIENTOS NO FUNCIONALES

El objetivo de esta actividad es servir como guía al resto de las fases que intervienen en el proceso de medición y evaluación de la calidad.

3.2.1. Definición de la Necesidad de Información

La *necesidad de información* para el presente caso de estudio es: “comprender y mejorar el sub-concepto de calidad externa *Accesibilidad* del Prototipo de Aplicación (Sistema) Web de Gestión de Guardias Médicas”.

- **Propósito:** comprender y mejorar.
- **Punto de vista de usuario:** usuarios con algún tipo de discapacidad parcial y/o personas mayores.
- **Concepto Calculable:** sub-concepto de calidad externa “Accesibilidad”, a su vez perteneciente al concepto “Usabilidad” planteado por el Modelo de Calidad de Producto Sistema/Software de ISO 25010. *Se considerarán aquellos atributos de Accesibilidad que permitan satisfacer el Principio Comprensible, Nivel A de WCAG 2.0*
- **Categoría de entidad:** Aplicación Web (Sistema).
- **Ente:** páginas y formularios web del Prototipo de Aplicación Web de Gestión de Guardias Médicas.

3.2.2. Selección de un Modelo de Conceptos

Otra actividad necesaria para la definición de Requerimientos No-Funcionales es la de “Seleccionar un Modelo de Conceptos”, el cual tiene como objetivo, dado un modelo, agregar o quitar conceptos, subconceptos y atributos además de especificar la forma en la cual se relacionan los mismos.

Se define como *concepto calculable*, a aquel que no puede ser medido directamente sino que su valor se obtiene en base a otros subconceptos y y/o atributos que lo definan; es decir que representan características de alto nivel de abstracción. Un subconcepto, es a su vez, un concepto

calculable. En cambio, los *atributos*, pueden ser medidos en forma directa mediante una métrica y constituyen las hojas del árbol de requerimientos.

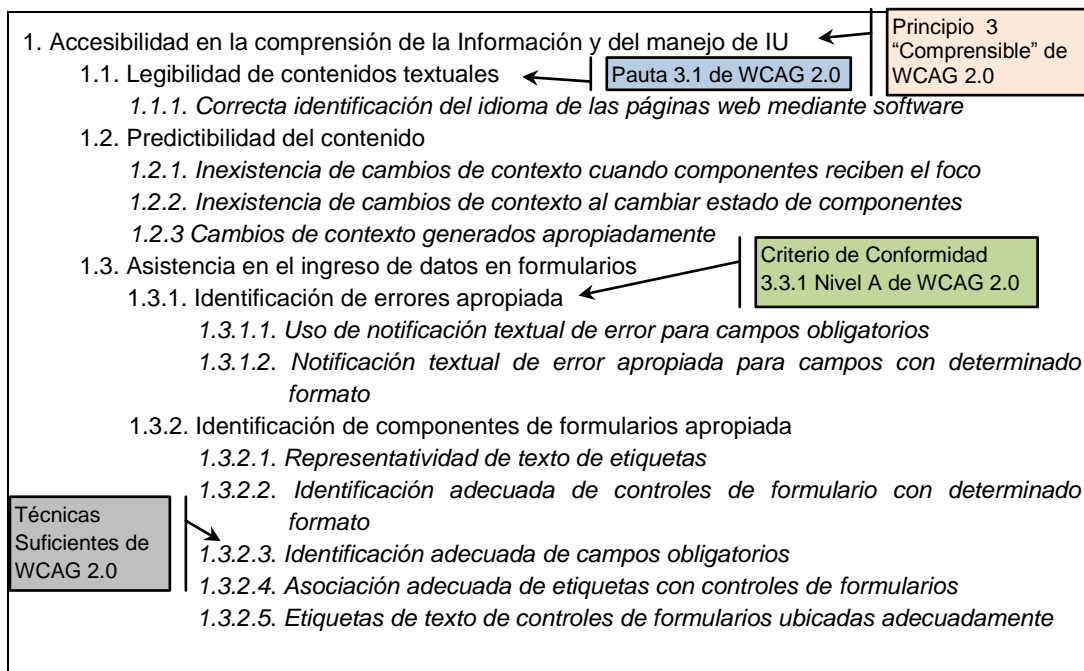
El Modelo de Calidad de Producto Sistema/Software de ISO 25010 que se consideró para este trabajo, solo define el sub-concepto Accesibilidad, sin especificar otros sub-conceptos o atributos que permitan evaluarla. Por ello en este trabajo se define un modelo que extiende al mencionado anteriormente, conformado por todos los sub-conceptos, atributos y relaciones, de manera que cumpla con la necesidad de información planteada.

Para la elaboración del modelo de conceptos se analizaron las “Pautas de Accesibilidad para el Contenido Web (WCAG) 2.0”, recomendación oficial de W3C. Se realizó una correlación entre sub-conceptos y atributos del modelo de conceptos y las pautas, principios, criterios de conformidad nivel A y técnicas suficientes de HTML de las WCAG 2.0. Como resultado de este estudio y análisis se planteó un árbol de requerimientos preliminar del concepto Accesibilidad presentado en (Gallardo & Quinteros, 2014). En la presente sección, se tomó como base el trabajo referenciado anteriormente y se completó la definición de requisitos no funcionales para satisfacer el “Principio Compresible” de WCAG 2.0.

De esta forma, el árbol de requerimientos (modelo de conceptos) obtenido, se estructura como sigue: el nivel 1 corresponde en nuestro caso al concepto Accesibilidad, más precisamente al Principio Compresible de WCAG 2.0. Luego los sub-conceptos del nivel 2 representan las pautas asociadas al mencionado principio; los sub-sub-conceptos del nivel 3 se adecuaron de acuerdo a los Criterios de conformidad Nivel A que se deben cumplir para satisfacer cada pauta, y finalmente, en el nivel 4 se especificaron los atributos que representan a las técnicas suficientes relacionadas con HTML para satisfacer los criterios de conformidad Nivel A.

El modelo de conceptos o árbol de requerimientos resultante es el que se muestra en la Tabla 2:

Tabla 2: Árbol de requerimientos para evaluar la “Accesibilidad en la comprensión” de una WebApp



3.2.2.1. Definiciones de conceptos y atributos del árbol de requerimientos:

- **Concepto Calculable: Accesibilidad en la comprensión de la Información y del manejo de IU**

La “Accesibilidad”, considerada como un concepto de alto nivel de abstracción, representa el grado en que el sistema pueda ser utilizado por personas con un rango más amplio de características y capacidades para lograr un objetivo específico en un contexto de uso específico (ISO/IEC 25010, 2011). En el presente caso de estudio, esta característica será evaluada por la presencia de propiedades del sistema que apoyan la *comprensibilidad de la información y del manejo de IU*, es decir, que el valor obtenido para este concepto representa el grado en que el sistema

proporciona información y manejo de interfaces de usuario de una manera entendible para los usuarios.

- **Sub-Concepto 1.1: Legibilidad de contenidos textuales**

Grado en que el sistema proporciona información o componentes textuales de una manera legible y comprensible.

- **Atributo 1.1.1: Correcta identificación del idioma de las páginas web mediante software**

Definición: el lenguaje humano del contenido de una página web debe poder ser determinado por software, a partir de datos suministrados por el autor, de manera que los diferentes agentes de usuario y tecnologías asistivas, puedan extraer y presentar con mayor precisión esta información a los usuarios en diferentes modalidades, los cuales estarán en mejores condiciones de comprender el contenido. Para ello, se debe especificar en el atributo “lang” y/o “xml:lang” del elemento <html> de cada página web, el código de idioma que identifica un lenguaje natural hablado, escrito o usado de cualquier modo para la comunicación de información entre personas. La especificación “BCP (Best Current Practice) N° 47: Etiquetas para la Identificación de Idiomas” (BCP 47: Tags for Identifying Languages, 2009) define y explica los códigos de idioma que deben ser usados en los documentos HTML, los cuales consisten en un código principal y una serie de sub-códigos opcionales:

`código-de-idioma = código-principal ("-" subcódigo) *`

A su vez, en el estándar ISO 639-2 (1998) se definen las abreviaturas para los lenguajes que se utilizan como código principal del idioma²².

²² Lista de códigos de Lenguajes en: http://www.loc.gov/standards/iso639-2/php/code_list.php

Objetivo: encontrar el grado en que se puede identificar correctamente el idioma predeterminado de las páginas web del sistema mediante el atributo “lang” del elemento <html>.

Documentos WCAG 2.0 relacionados:

- ↪ Criterio de Conformidad 3.1.1 – Lenguaje de la Página.
- ↪ Técnica Suficiente H57 – Uso de atributos de lenguaje en el elemento html.

▪ **Sub-Concepto 1.2: Predictibilidad del contenido**

Grado en que el contenido del sistema se muestra y opera de una manera predecible o esperada por el usuario.

▪ **Atributo 1.2.1: Inexistencia de cambios de contexto cuando componentes reciben el foco**

Definición: para asegurar que la funcionalidad y navegación de una página sea predecible, cualquier componente que recibe el foco no debe desencadenar ningún cambio de contexto en la misma. La palabra “componente” se refiere a un elemento de interfaz de usuario como ser: campo de entrada de datos, botón radio, botón de envío, etc.

Los *cambios de contexto*, representan aquellas modificaciones importantes en el contenido de la página Web que si se producen sin el conocimiento del usuario, pueden desorientar a aquellos que no son capaces de ver la página completa de manera simultánea. Cambios de contexto incluyen cambios de: *agentes de usuario* (navegador, reproductores multimedia, plugins, etc); *vistas* (ventanas, marcos); *foco a otro elemento*; contenido que cambia el significado de la página.

Cabe aclarar, que un *cambio de contenido* no es siempre un cambio de contexto. Por ejemplo, un menú dinámico, o un control de pestañas (o fichas) no cambian necesariamente el contexto. Se recomienda que todos

los cambios de contexto sucedan solo por una acción específica del usuario, como por ejemplo: hacer un clic en un link o presionar un botón de envío de datos y no cuando un elemento reciba el foco.

Objetivo: encontrar el grado de inexistencia de cambios de contexto automáticos o sin consentimiento del usuario cuando elementos de las páginas reciben el foco.

Documentos WCAG 2.0 relacionados:

- ↳ Criterio de Conformidad 3.2.1 – Al recibir el foco.
- ↳ Técnica Suficiente G107 – Uso de “activaciones” en lugar del “foco” de elementos como disparadores para cambios de contexto.

▪ **Atributo 1.2.2: Inexistencia de cambios de contexto al cambiar estado de componentes**

Definición: el sistema debe evitar provocar cambios de contextos automáticos y sin conocimiento del usuario, cuando se producen modificaciones en el estado de los elementos de interfaz de usuario. Se consideran cambios de estado de componentes cuando por ejemplo: se elige una opción en un control de casilla de verificación o botón de radio, se ingresan datos dentro de un campo de texto, o se cambia la opción seleccionada en una lista desplegable. En cambio, al presionar un enlace o seleccionar una pestaña (o ficha) en un control de fichas, se está activando el control, y no cambiando su estado.

Esta práctica beneficia a los usuarios con discapacidades haciendo el contenido más predecible, ya que cambios de contextos inesperados pueden desorientar a usuarios con discapacidad visual o con limitaciones cognitivas.

Objetivo: determinar el grado de inexistencia de cambios de contexto automáticos o sin consentimiento del usuario, cuando se modifica el estado de componentes de interfaz de usuario.

Documentos WCAG 2.0 relacionados:

↳ Criterio de Conformidad 3.2.2 – Al recibir entradas de datos.

▪ **Atributo 1.2.3: Cambios de contexto generados apropiadamente**

Definición: los cambios de contexto se deben activar sólo mediante una acción específica por parte del usuario, o en caso contrario, se debe informar y advertir correctamente al usuario del comportamiento de los componentes de interfaz de usuario y/o de la página web.

Dado que con el uso de un botón o un link se estarían activando tales controles y el usuario usualmente espera algún cambio en el contenido o comportamiento, se trata de controles adecuados para provocar explícitamente un cambio de contexto predecible y apropiado, si éste fuera necesario, y es una práctica que no crea confusión entre los usuarios.

Por otra parte, en el caso que no se pueda cumplir con la sugerencia anterior, se deben proporcionar instrucciones e información a los usuarios acerca de lo que ocurrirá cuando se modifique el estado de los componentes de un formulario, de tal manera que el usuario sea consciente de antemano de los cambios que se producirán.

Situación de ejemplo: en un proceso multi-etapa en la que los usuarios deben completar una serie de campos a través de diferentes páginas (o pestañas), y donde los cambios de estado de elementos de interfaz de usuarios causarían un cambio de contexto, se proporciona una instrucción como parte del proceso antes de la fase o página donde se producirá el cambio de contexto.

Objetivo: encontrar el grado en que el sistema inicia cambios de contexto apropiados, mediante alguna de las siguientes formas:

- utilizando botones (`<input type="submit">`, `<input type="image">`, `<input type="button">`, o `<button type="submit">`)

- utilizando enlaces ()
- proporcionando instrucciones e información antes que se produzca el cambio de contexto.

Documentos WCAG 2.0 relacionados:

- ↪ Criterio de Conformidad 3.2.2 – Al recibir entradas de datos.
- ↪ Técnica Suficiente G80 – Proporcionar un botón submit para iniciar un cambio de contexto.
- ↪ Técnica Suficiente H32 - Proporcionar botones submit
- ↪ Técnica Suficiente G13: Describir los cambios de contexto que se producirán al modificar estado de componentes.

- **Sub-Concepto 1.3: Asistencia en el ingreso de datos en formularios**

Grado en que el sistema ayuda al usuario a evitar y corregir errores cuando ingresa datos mediante formularios web.

- **Sub-Concepto 1.3.1: Identificación de errores apropiada**

Grado en que el sistema proporciona una orientación clara y específica sobre un error producido por alguna información suministrada por el usuario mediante un formulario web y que no ha sido aceptada por el sistema por no cumplir con determinados requisitos, de tal manera que el usuario pueda comprender y corregir el error.

- **Atributo 1.3.1.1: Uso de notificación textual de error para campos obligatorios**

Definición: cuando el sistema presenta un formulario para el ingreso de datos por parte del usuario y el mismo contiene campos obligatorios, se debe informar al usuario mediante una descripción de texto en el caso de que no se hayan completado dichos campos obligatorios para identificar

cuál o cuáles campos se omitieron. Si bien se pueden utilizar otros mecanismos como indicar con color el campo donde se produjo el error, es importante que exista una notificación textual para que sea accesible a todos los usuarios y tecnologías de asistencia.

Objetivo: encontrar el grado en que el sistema, para todos los formularios del caso de estudio, notifica al usuario mediante una descripción de texto, sobre un error producido cuando dicho usuario envía los datos de un formulario sin haber completado cada campo indicado como obligatorio.

Documentos WCAG 2.0 relacionados:

- ↳ Criterio de Conformidad 3.3.1 – Identificación de errores.
- ↳ Técnica Suficiente G83 – Proporcionar descripciones de texto para identificar campos obligatorios que no han sido completados.

▪ **Atributo 1.3.1.2: Notificación textual de error apropiada para campos con determinado formato**

Definición: cuando el sistema presenta un formulario para el ingreso de datos por parte del usuario y el mismo contiene campos que requieren un formato o valor en particular, se debe proporcionar una adecuada ayuda para la corrección de errores de entrada cuando la información suministrada por el usuario se valida y detecta como incorrecta. Se proporciona en este caso, una notificación al usuario mediante una descripción de texto para indicar la naturaleza y ubicación (campo) del error de entrada. Si bien se pueden utilizar otros mecanismos como indicar con color el campo donde se produjo el error, es importante que exista una notificación textual para que sea accesible a todos los usuarios y tecnologías de asistencia.

Objetivo: Encontrar el grado en que el sistema, para todos los formularios del caso de estudio, notifica al usuario mediante una descripción de texto apropiada, sobre un error producido cuando dicho usuario envía los datos de un formulario con un valor o formato incorrecto en algún campo. La

notificación debe incluir la identificación del campo donde se produjo el error y proporcionar información sobre la naturaleza de la entrada inválida y cómo solucionarlo.

Documentos WCAG 2.0 relacionados:

- ↪ Criterio de Conformidad 3.3.1 – Identificación de errores.
- ↪ Técnica Suficiente G84 - Proporcionar una descripción de texto cuando el usuario ingresa datos que no están en la lista de valores permitidos.
- ↪ Técnica Suficiente G85 - Proporcionar una descripción de texto cuando la entrada del usuario no corresponde con el formato o los valores requeridos.

▪ **Sub-Concepto 1.3.2: Identificación de componentes de formularios apropiada**

Grado en que el sistema proporciona instrucciones o etiquetas que identifican los controles o componentes interactivos de un formulario web para informar al usuario sobre la manera correcta o esperada para completar un campo o también para incluir cualquier otra información adicional. Se consideran componentes interactivos de un formulario tanto los elementos que permiten el ingreso de datos como campos de entrada (input), casilla de verificación (checkbox), botón de radio, etc. como también aquellos que permiten interactuar con el mismo como los botones por ejemplo.

▪ **Atributo 1.3.2.1: Representatividad de texto de etiquetas**

Definición: cada etiqueta de texto debe indicar claramente el propósito del componente interactivo de formulario al cual representa, ya sean campos de entradas o botones o cualquier otro control del formulario.

Objetivo: determinar en qué medida el sistema proporciona un texto de etiqueta representativo para los componentes que la posean, de todos los formularios web del caso de estudio.

Documentos WCAG 2.0 relacionados:

- ↳ Criterio de Conformidad 3.3.2 - Etiquetas o instrucciones.
- ↳ Técnica Suficiente G131 – Proporcionar etiquetas descriptivas.

▪ **Atributo 1.3.2.2: Identificación adecuada de controles de formulario con determinado formato**

Definición: cuando se requiere el ingreso de algún dato, mediante componentes de un formulario y con un determinado formato, se puede ayudar al usuario a evitar errores de entrada, a través de una descripción textual, primero indicando que tal campo requiere una entrada específica y luego informándoles de las características o un ejemplo del formato de los datos que se deben introducir.

Objetivo: determinar en qué medida el sistema proporciona adecuadamente indicaciones e instrucciones de texto para identificar y brindar información sobre el formato requerido de un campo de entrada en particular, dentro de todos los formularios del caso de estudio.

Documentos WCAG 2.0 relacionados:

- ↳ Criterio de Conformidad 3.3.2 - Etiquetas o instrucciones.
- ↳ Técnica Suficiente G89 – Proporcionar instrucciones y ejemplos del formato de datos esperado.

▪ **Atributo 1.3.2.3: Identificación adecuada de campos obligatorios**

Definición: es necesario proporcionar una clara indicación cuando se requiere un control de formulario específico en un formulario web para que se ingresen los datos con éxito. Esta identificación puede realizarse ya sea mediante símbolos, imagen o texto que indique adecuadamente el campo obligatorio. En el caso de utilizar símbolos o imágenes, el usuario

debe ser informado de su significado antes de la primera utilización de los mismos. Cuando la identificación del campo obligatorio se realice mediante texto, el mismo debe ser incluido dentro de los elementos <label> o <legend> del formulario web, según se trate de un campo simple como un <input> o de un grupo de controles de formularios (botón de radio o casilla verificación) respectivamente, con el fin de ayudar a la accesibilidad de personas discapacitadas mediante la tecnologías asistivas, ya que las mismas utilizan estos elementos para brindar la información correspondiente.

Objetivo: determinar en qué medida el sistema proporciona adecuadamente, dentro de todos los formularios del caso de estudio, una identificación por cada campo de entrada obligatorio, ya sea a través de texto incluido dentro de los elementos <label> o <legend> del formulario web, o símbolos o imágenes, previa notificación del significado para los últimos dos casos.

Documentos WCAG 2.0 relacionados:

- ↳ Criterio de Conformidad 3.3.2 - Etiquetas o instrucciones.
- ↳ Técnica Suficiente H90 – Identificar controles de formularios obligatorios utilizando la etiqueta label o legend.

▪ **Atributo 1.3.2.4: Asociación adecuada de etiquetas con controles de formularios**

Definición: dentro de la definición de un formulario web, se puede asociar explícitamente un control de formulario con una etiqueta de texto, incluyendo dicho texto dentro del elemento <label> y con un valor del atributo “for” del elemento que coincida con el valor del atributo “id” del control de formulario. A través de este mecanismo de asociación, se proporciona asistencia a todos los usuarios que necesitan ayuda para entender el propósito del campo. Por otra parte, se puede contar con un área seleccionable más amplia para el control del formulario, ya que al

hacer clic en la etiqueta automáticamente el foco se posiciona en el control, esto puede ser útil para usuarios con control motriz deteriorado. Hay que tener en cuenta que sólo los siguientes elementos de formularios web pueden utilizar el elemento <label> asociado: <input type="text">, <input type="checkbox">, <input type="radio">, <input type="file">, <input type="password">, <textarea>, <select>

Objetivo: determinar el grado en que el sistema realiza una adecuada asociación, entre etiquetas de texto y los controles de formularios mediante el elemento <label> y uso del atributo "for", dentro de todos los formularios del caso de estudio.

Documentos WCAG 2.0 relacionados:

- ↳ Criterio de Conformidad 3.3.2 - Etiquetas o instrucciones.
- ↳ Técnica Suficiente H44 – Uso de elementos label para asociar etiquetas de texto con controles de formularios.

▪ **Atributo 1.3.2.5: Etiquetas de texto de controles de formularios ubicadas adecuadamente**

Definición: la ubicación apropiada de las etiquetas correspondientes a los campos de formulario ayuda al usuario a comprender más fácilmente formularios complejos y a localizar campos específicos. Para una visualización predecible, en un idioma de lectura de izquierda a derecha, las etiquetas deben situarse a la izquierda o por encima de los campos del formulario, y en el caso de botones de opción y casillas de verificación se deben colocar después de los mismos.

Este posicionamiento de etiquetas beneficia a la accesibilidad porque cuando se colocan antes de los campos de entrada, permite la alineación de las etiquetas, siendo más fácil localizarlas con un amplificador de pantalla, o encontrarlas en un diseño de formulario de columna vertical. Por último, si el campo tiene datos en él, es más fácil de entender o

comprobar los datos si se lee la etiqueta y después el contenido en lugar de lo contrario.

Por otra parte, las casillas de verificación y botones de radio tienen un ancho uniforme, mientras que las etiquetas pueden variar en longitud. Entonces ubicar un botón de opción o casilla de verificación en primer lugar, permite que tanto los botones como las etiquetas se alineen verticalmente.

Objetivo: determinar el grado en que el sistema, dentro de todos los formularios del caso de estudio, dispone adecuadamente las etiquetas de texto dependiendo del tipo de control de formulario que se trate.

Documentos WCAG 2.0 relacionados:

- ↳ Criterio de Conformidad 3.3.2 - Etiquetas o instrucciones.
- ↳ Técnica Suficiente G162 - Posicionamiento de etiquetas para maximizar la previsibilidad de las relaciones.

3.3. (A2) DISEÑO DE LA MEDICIÓN

Siguiendo el proceso definido en GOCAME, esta segunda actividad consiste en seleccionar la métrica más significativa desde un repositorio de métricas para cada atributo del árbol de requerimientos definido en la etapa anterior.

Para el presente caso de estudio, resulta necesario definir cada una de las métricas utilizadas, debido a que no existe un repositorio de métricas ya diseñadas que se ajusten exactamente a los requerimientos planteados.

Las métricas propuestas en este trabajo han sido definidas para cumplir con las técnicas suficientes del estándar WCAG 2.0, es decir, reflejan las formas confiables de satisfacer los criterios de conformidad de la norma. En este sentido, por ejemplo para el “*Atributo 1.3.1.1: Notificación textual*”

de error apropiada para campos obligatorios”, hubiese bastado con definir una función binaria (100=cumple/0=no cumple). Sin embargo, dada la riqueza del método creemos oportuno hacer uso de ella, habiendo, en este caso, optado por la definición de una métrica que aplica una escala de proporción, que devuelve valores entre 0 y 100, donde 0 corresponde a la ausencia total de satisfacción del criterio, pasando por valores intermedios de satisfacción hasta llegar a 100, valor que corresponde con el cumplimiento total del atributo del árbol de requerimientos. La validación de dicha métrica, al igual que las restantes métricas indirectas, ha sido realizada de forma teórica, de acuerdo al marco de validación propuesto por (Olsina, 1999). En cuanto a las métricas directas, cuando se trata de métricas objetivas, no es necesaria su validación. Solo en el caso de métricas directas subjetivas, el valor final surgirá del consenso de dos o más evaluadores expertos, con el fin de minimizar errores de valoración intencionales y/o involuntarios. Esto último también podría hacerse de forma empírica, es decir, por medio de la observación y la planificación de experimentos y encuestas.

Por cuestiones de legibilidad no se describen todas las métricas en este apartado, sólo se exponen a modo de ejemplo las métricas correspondientes a los Atributos *“1.1.1 - Correcta identificación del idioma de las páginas web mediante software”* y *“1.3.2.2 - Identificación adecuada de controles de formulario con determinado formato”*, las cuales se pueden observar en la Tabla 3 y Tabla 4 respectivamente. La definición completa de las métricas diseñadas se puede consultar en el Anexo A.

Tabla 3: Definición de métricas para el Atributo 1.1.1 del modelo de requerimientos

Atributo 1.1.1. Correcta identificación del idioma de las páginas web mediante software			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p><i>Nombre:</i> Porcentaje de páginas web²³ con identificación de idioma mediante software (%PIHTML)</p> <p><i>Tipo:</i> Indirecta</p> <p><i>Métricas relacionadas:</i> 1) Número de páginas web que poseen el atributo lang dentro del elemento html (#PALHTML) 2) Total de páginas web (#TPW)</p>	<p>Determinar el porcentaje entre el número de páginas web que posean una identificación del idioma mediante el atributo lang dentro del elemento html y el número total de páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%PIHTML = \frac{\#PALHTML}{\#TPW} * 100$	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %
<p><i>Nombre:</i> Número de páginas web que poseen el atributo lang dentro del elemento html (#PALHTML)</p> <p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Páginas web con identificación de idioma mediante atributo lang del elemento html</p>	<p>Determinar el número de páginas web que posean una identificación del idioma mediante atributo lang dentro del elemento html.</p>	<p>Especificación de método de medición: Inicializar la variable #PALHTML = 0. Por cada documento html diferente generado por la aplicación web, dentro del Rol de Supervisor, verificar que:</p> <ol style="list-style-type: none"> 1) el elemento "html" contiene el atributo "lang" y/o el atributo "xml:lang" 2) el valor de lang es válido y corresponde a algún idioma en particular según la codificación establecida en ISO 639-2 y BCP 47. 3) el lenguaje del texto en general de la página web coincide con el código del lenguaje especificado. <p>Si se cumplen todas las condiciones anteriores, entonces #PALHTML = #PALHTML + 1</p> <p>Si no se cumplen alguna de las condiciones anteriores, no sumar ningún valor a #PALHTML.</p> <p>Tipo de método de medición: objetivo</p>	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: páginas web con identificación de idioma
<p><i>Nombre:</i> Total de páginas web (#TPW)</p> <p><i>Tipo:</i> Directa</p> <p>Atributo relacionado:</p>	<p>Determinar el número total de páginas (documentos html) de la aplicación web.</p>	<p>Especificación de método de medición: Inicializar la variable #TPW = 0. Ingresar a la URL principal de la webapp, e identificarse con el Rol de usuario "Supervisor". Luego navegar</p>	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala:

²³ Una página web es la representación visual de un documento HTML, por lo que mediante un navegador web se puede ver el código fuente de la misma.

páginas web de la WebApp		<p>por todas opciones de menú y otras opciones que provea la aplicación, agregando 1 (uno) a #TPW por cada página web diferente (documento html) generada.</p> <p>Nota: No se sumarán las páginas correspondientes a paginación de resultados.</p> <p>Tipo de método de medición: objetivo</p>	<p>Absoluta</p> <p>▪ Unidad: Página web</p>
--------------------------	--	---	--

Tabla 4: Definición de métricas para el Atributo 1.3.2.2 del modelo de requerimientos

Atributo 1.3.2.2. Identificación adecuada de controles de formulario con determinado formato			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p><i>Nombre:</i> Porcentaje de identificación adecuada para campos con formato (%IACF)</p> <p><i>Tipo:</i> Indirecta</p> <p><i>Métricas relacionadas:</i></p> <p>1) Numero de identificaciones adecuadas para campos con formato (#IACF)</p> <p>2) Total de campos con formato</p>	<p>Determinar el porcentaje entre el número de identificaciones adecuadas para campos con formato y el número total de campos con un formato requerido, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%IACF = \frac{\#IACF}{\#CF} * 100$ <p>Si #CF = 0 entonces %IACF = 100 ya que no se considera como una situación de incumplimiento sino de inexistencia de campos con formato.</p>	<p>▪ Representación: Continua</p> <p>▪ Tipo de valor: Real</p> <p>▪ Tipo de escala: Proporción</p> <p>▪ Unidad: Porcentaje</p> <p>▪ Acrónimo: %</p>
<p><i>Nombre:</i> Numero de identificaciones adecuadas para campos con formato (#IACF)</p> <p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Indicaciones adecuadas para campos con formato específico</p>	<p>Determinar el número de identificaciones adecuadas para campos que requieran un formato específico correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de método de medición: Inicializar la variable #IACF = 0. Por cada página web de la aplicación, y a su vez por cada formulario, reconocer los controles de formulario que sólo aceptan los datos de entrada del usuario en un formato determinado. Agregar 1 (uno) a #IACF por cada uno de los controles de formulario observados anteriormente que posean una clara identificación e información adecuada sobre el formato esperado.</p> <p>Tipo de método de medición: subjetivo</p>	<p>▪ Representación: Discreta</p> <p>▪ Tipo de valor: Entero</p> <p>▪ Tipo de escala: Absoluta</p> <p>▪ Unidad: indicaciones adecuadas para campos con formato</p>
<p><i>Nombre:</i> Total de campos con formato (#CF)</p>	<p>Determinar el número de campos que posean una indicación de un valor o</p>	<p>Especificación de método de medición: Inicializar la variable #CF=0. Por cada página web, y a su</p>	<p>▪ Representación: Discreta</p>

<p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Campos que requieren un determinado formato</p>	<p>formato requerido correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>vez por cada formulario, completar el mismo ingresando datos erróneos en todos los campos que posean una indicación del formato o valor requerido o que se infiera que puede requerir un formato particular.</p> <p>Agregar 1 (uno) a #CF por cada notificación que emita el sistema, ya sea textual o no, informando que los datos ingresados son incorrectos y que requieren un formato específico.</p> <p>Tipo de método de medición: subjetivo</p>	<p>▪ Tipo de valor: Entero</p> <p>▪ Tipo de escala: Absoluta</p> <p>▪ Unidad: Campo con formato específico</p>
---	---	--	---

3.4. (A4) DISEÑO DE LA EVALUACIÓN

En esta etapa se define la evaluación mediante el uso de indicadores, los cuales especifican cómo interpretar los valores de los atributos (resultados de métricas) y de los conceptos de más alto nivel, permitiendo conocer de este modo el grado de satisfacción de todos los requerimientos para una necesidad de información dada.

Se deben identificar aquí dos tipos de indicadores:

- *Indicador elemental:* el cual interpreta el valor de un atributo mediante el uso de un modelo elemental
- *Indicador global:* permite evaluar un concepto de alto nivel de abstracción (concepto calculable de diferentes niveles de abstracción), y su valor se deriva de otros indicadores haciendo uso de un modelo global.

Todos los modelos planteados tienen asociados criterios de decisión.

3.4.1. Identificación de Indicadores elementales

La especificación del proceso de M&E GOCAME indica que se debe identificar un indicador elemental para cada uno de los atributos

determinados en el árbol de requerimientos, usando como entrada la métrica asociada en la especificación de métricas.

Los valores obtenidos a partir del modelo elemental del indicador planteado, serán números entre 0 y 100, indicando con esto el porcentaje de satisfacción cubierto para cada atributo. Para nuestro análisis, la correspondencia entre métrica e indicador elemental es directa, ya que todas las métricas definidas poseen una escala porcentual que determina el nivel de cumplimiento del requisito. Esto es posible porque tanto la métrica y el indicador, poseen la misma escala y unidad.

A continuación, se muestra en la Tabla 5 la definición genérica del indicador que se aplicará a todas las métricas de atributos:

Tabla 5: Definición de Indicador Elemental

Nombre Indicador Elemental	Especificación Modelo Elemental	Criterio de Decisión	Escala Numérica
GCA: Grado de cumplimiento del Atributo(j) j = atributos del árbol de requerimientos	$GCA(j) = \text{Valor Metrica}(j)$ j = atributos del árbol de requerimientos	<p>Niveles de aceptabilidad:</p> <ul style="list-style-type: none"> ▪ Insatisfactorio. Indica que acciones de cambio deben tomarse con una alta prioridad. Rango: $0 \leq GCA(j) \leq 50$ ▪ Regular. Indica una necesidad de acciones de mejora. Rango: $50 < GCA(j) \leq 75$ ▪ Satisfactorio. Indica una calidad satisfactoria de la característica analizada. Rango: $75 < GCA(j) \leq 100$ 	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %

3.4.2. Identificación de Indicadores globales

Para definir el modelo de agregación de los valores de indicadores parciales y globales se aplica el *Modelo de Puntaje de Preferencia Lógica (LSP)*, el cual es un método cuantitativo basado en técnicas de puntuación y lógica continua de preferencias propuesto por Dujmovic (Dujmovic, 1996) y que ha sido incorporado por la metodología WebQEM, la que a su vez forma parte de la estrategia GOCAME. Dicho modelo se

define por la siguiente función de agregación general denominada “Media de Potencia Pesada”, tal como se especifica en la ecuación (1):

$$IG = (W_1 * I_1^r + W_2 * I_2^r + \dots + W_j * I_j^r)^{1/r} \quad (1)$$

donde:

- ↪ **IG** representa el indicador parcial o global a ser calculado.
- ↪ **I_i** son los valores de los indicadores del nivel más bajo, $0 \leq I_i \leq 100$
- ↪ **W_i** representa los pesos, $(W_1 + W_2 + \dots + W_j) = 1$; $W_i > 0$; $W_i = 1/j$; $i=1\dots j$ donde j es el número total de indicadores. La asignación de los pesos correspondientes a los atributos y subconceptos es equivalente, porque se considera que todos los atributos o subconceptos en un mismo nivel del árbol, tienen igual importancia.
- ↪ **r** representa una función de agregación y varía entre $-\infty \leq r \leq \infty$. Para nuestro caso de estudio, se aplicará la función de conjunción/disyunción generalizada denominada “Media Aritmética”, con lo cual **r=1**, convirtiendo a la ecuación (1) en la ecuación (2):

$$IG = (W_1 * I_1 + W_2 * I_2 + \dots + W_j * I_j) \quad (2)$$

El Criterio de decisión a utilizar será el mismo que el definido para los indicadores elementales:

- **Insatisfactorio.** Indica que acciones de cambio deben tomarse con una alta prioridad. *Rango:* $0 \leq GCA(j) \leq 50$
- **Regular.** Indica una necesidad de acciones de mejora. *Rango:* $50 < GCA(j) \leq 75$
- **Satisfactorio.** Indica una calidad satisfactoria de la característica analizada. *Rango:* $75 < GCA(j) \leq 100$

CAPITULO IV - MODELADO Y DISEÑO DEL PROTOTIPO DE APLICACIÓN WEB

4.1. INTRODUCCION

En el presente capítulo se expondrán en diferentes secciones, la Especificación de Requisitos de Software (ERS), el cual es un documento donde se plasma el resultado del análisis y relevamiento realizado en el contexto de las guardias médicas y donde se especifican los requisitos funcionales del sistema web propuesto.

Posteriormente se presenta el modelado de dichos requisitos funcionales mediante Casos de Uso y los modelos resultantes luego de aplicar el enfoque UWE en el diseño del prototipo de Aplicación Web propuesto.

Cabe destacar que tanto la Especificación de Requisitos de Software, como el modelado del prototipo propuesto en este Trabajo Final, constituyen una refinación y extensión del trabajo presentado en (Gallardo & Ahumada, 2015), donde se expusieron los artefactos resultantes de las etapas tempranas del desarrollo del prototipo propuesto.

4.2. ESPECIFICACION DE REQUISITOS DE SOFTWARE

4.2.1. Visión General del Documento

Se proporcionará en este apartado, una vista general de la Especificación de Requisitos de Software (ERS), donde se mencionará el objetivo, alcance, definiciones y acrónimos y las referencias que correspondan.

4.2.1.1. Propósito

La Especificación de Requisitos de Software tiene por objetivo conseguir una estructura bien definida y un documento bien formado de las funcionalidades y restricciones de la Aplicación Web a desarrollar.

El público al que está dirigido es principalmente el equipo de la Dirección de Informática y Comunicación Social, dependiente del Ministerio de Salud de la Provincia de Catamarca, quienes serán los encargados del mantenimiento y actualización del sistema en mención. Así también, la presente ERS se pondrá a disposición de los directivos de las Áreas involucradas en la gestión y control de las Guardias Médicas en el ámbito provincial.

4.2.1.2. Alcance o Ámbito del Sistema

El nombre correspondiente al producto software a desarrollar es “Sistema Web de Gestión de Guardias Médicas”, en adelante SiWebGM.

El prototipo de aplicación web propuesto comprenderá la gestión de guardias médicas que se ofrecen en los distintos establecimientos asistenciales dependientes del Ministerio de Salud del Gobierno de la Provincia de Catamarca.

La aplicación web a desarrollar formará parte del “Sistema Administrativo de Guardias”, donde será necesaria la interacción con otros sistemas de software existentes, los cuales se mencionarán en detalle en la sección: *“4.2.2.2. Perspectiva de la Aplicación”*.

SiWebGM no cubrirá el proceso de liquidación de haberes por Guardias Médicas ni tampoco la gestión de Empleados y Contratos por Guardias.

4.2.1.3. Personal involucrado

Por cuestiones de privacidad, no se brindarán datos de personas físicas pero si se identificarán los roles involucrados en el desarrollo del sistema.

Rol	Jefe de Guardia de Área o Establecimiento Sanitario
Responsabilidades	Brindar información respecto al manejo diario de las guardias realizadas, para identificar aspectos relevantes a ser tenidos en cuenta

Rol	Personal de Área de Control de Guardias
Responsabilidades	Brindar información sobre las restricciones y controles que se deben realizar en el contexto del sistema de guardias

Rol	Personal de Área de Liquidación de Guardias
Responsabilidades	Brindar información referida a los datos que debe administrar y proporcionar la WebApp para la posterior liquidación de las guardias realizadas

Rol	Personal de Área de Recursos Humanos
Responsabilidades	Brindar información sobre los datos y relaciones que debe administrar la WebApp en relación con los legajos y contratos de guardias

4.2.1.4. Definiciones, Acrónimos y Abreviaturas

DEFINICIONES

Empleado de guardia: persona autorizada mediante un Contrato de guardia para atender las emergencias médicas, y que además posee el perfil y capacitación profesional conforme a las actividades específicas establecidas para el cumplimiento del servicio requerido. Puede tener varios contratos en diferentes Establecimientos Asistenciales (Áreas), participar en guardias de distintos Servicios Internos de Área compatibles con su especialidad, y también poseer un cargo en la Planta Pública

Permanente de la Provincia siempre y cuando no exista compatibilidad horaria.

Guardia Médica: servicio prestado por un empleado de guardia (que posee contrato vigente en un Área en particular) para atender emergencias médicas durante un horario estipulado y un Área específica. Cada guardia médica está asociada a un Periodo de Liquidación y posee entre otras, dos propiedades (una por cada categoría) de acuerdo a la siguiente clasificación (Ver Glosario para más detalle):

- Según el tipo de contrato del empleado que realiza la guardia: **profesional o técnica.**
- Según el tipo de permanencia en el lugar de trabajo de la guardia: **activa o pasiva.**

Establecimiento Asistencial: también denominado *Área*, se refiere a la Institución donde se realizan las guardias. Pertenece a un Servicio Administrativo Financiero.

Servicio Administrativo Financiero (SAF): Organismo que agrupa varias Áreas y que puede encargarse de la gestión y supervisión de guardias médicas de todas o algunas de sus Áreas.

Periodo de Liquidación de Área: se refiere a un periodo mensual (mes y año calendario) relacionado a un Área en particular, que posee un Identificador Único para futuras referencias en el proceso de liquidación. Cuando se registra una guardia, debe asignarse a un Periodo de Liquidación de acuerdo al Área del Contrato correspondiente.

↪ **ACRONIMOS**

SiWebGM: Sistema Web de Gestión de Guardias Médicas

SICOGM: Sistema de Gestión de Contratos por Guardias Médicas

SILIQGM: Sistema de Liquidación de Haberes por Guardias Médicas

SILQUIP: Sistema de Liquidación de Haberes Provincial

CUIL: Clave Única de Identificación Laboral

ABREVIATURAS

RN: Regla de negocio

AGM: Administrador de Guardias Médicas

SAF: Servicio Administrativo Financiero

CU: Caso de Uso

WebApp: Aplicación Web

4.2.1.5. Referencias

El presente documento de especificación de requisitos se basa en el estándar IEEE Std-830-1998 (IEEE Recommended Practice for Software Requirements Specifications)

4.2.1.6. Resumen

La documentación que a posteriori se transcribe, contiene los lineamientos generales de la Aplicación Web de Guardias Médicas y sus requerimientos tanto funcionales como no funcionales.

La organización del documento está compuesta de un primer acercamiento al sistema a través de su descripción general donde se trata la perspectiva, funciones, características de los usuarios, restricciones y dependencias. Luego se abordan los requerimientos específicos de la aplicación y para finalizar, se expone un glosario de los términos utilizados en el presente informe.

4.2.2. Descripción General de la Aplicación Web

4.2.2.1. Perspectiva de la Aplicación

El prototipo SiWebGM, formará parte de un conjunto de sistemas, que considerados integralmente, permiten resolver todos los aspectos y necesidades del “Sistema Administrativo de Guardias”. En la Figura 27 se muestra un diagrama donde se sitúa el Sistema Web propuesto y sus relaciones con los demás sistemas:

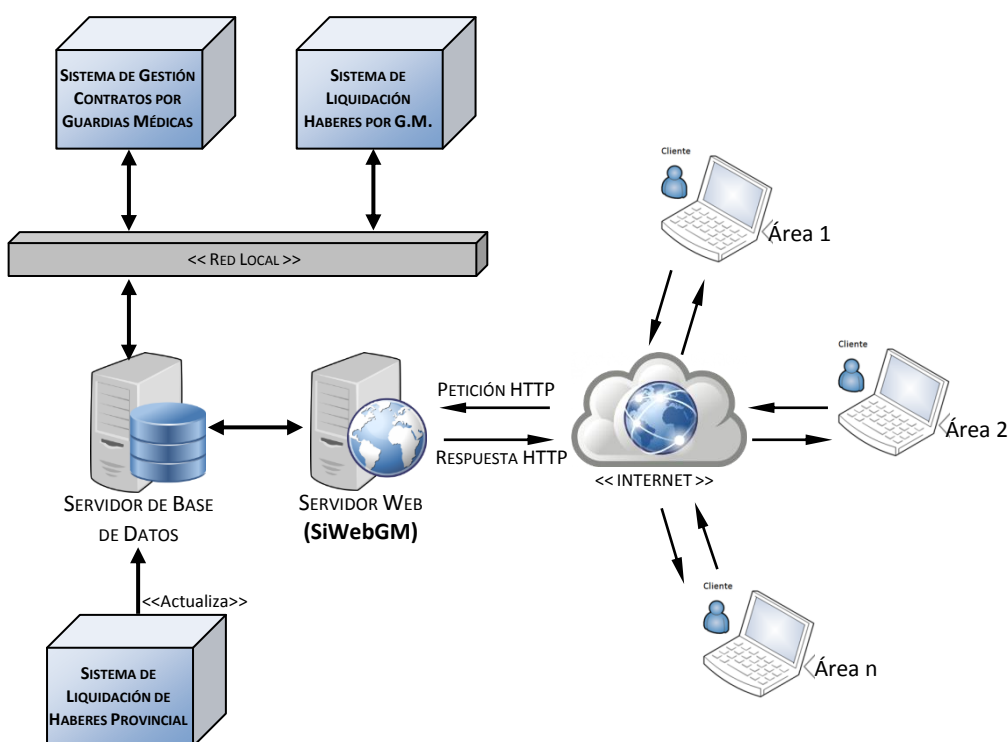


Figura 27: Interrelación de SiWebGM con otros sistemas software

A continuación se describen las interacciones de SiWebGM con los otros productos.

Interacción con el Sistema de Gestión de Contratos por Guardias Médicas (SICOGM): la Aplicación Web propuesta, se relaciona con el Sistema de Gestión de Contratos por Guardias Médicas, el cual es un sistema software de escritorio existente, cuya función principal es

administrar todos los datos relacionados a los empleados y a los contratos que se celebran entre las diferentes Áreas del Gobierno de la Provincia y un agente en particular. Cabe destacar que un mismo agente puede tener contratos de guardia vigentes en más de una Área.

El SiWebGM debe permitir la gestión de los registros de las guardias médicas realizadas, sólo de los empleados que posean un contrato vigente en algún Área del Gobierno, por lo cual debe acceder a la misma Base de Datos del Sistema de Contratos por Guardias Médicas.

El sistema SICOGM también se encarga de la gestión de los Servicios Administrativos Financieros y de las Áreas dependientes de éstos.

Interacción con el Sistema de Liquidación de Haberes por Guardias Médicas (SILIQGM): el prototipo web propuesto se debe relacionar también con el Sistema de Liquidación de Haberes por Guardias Médicas, compartiendo y accediendo a la misma Base de Datos. Los registros de las guardias realizadas que son gestionados por SiWebGM, serán tomados por el SILIQGM para efectuar el proceso de liquidación de haberes de las guardias por cada contrato que posea el empleado, además de otros conceptos propios de la liquidación. Desde SiWebGM se permitirá la gestión de los distintos registros de Periodos de Liquidación por cada Periodo y Área, de tal forma que al momento de cargar una guardia, éstas se relacionan con su N° de Periodo de Liquidación correspondiente para posteriores referencias en el proceso de liquidación. Por otra parte, cuando se realice el proceso de liquidación desde el SILIQGM, se debe cerrar dicho Periodo de Liquidación, bloqueando para el mismo, la modificación, eliminación o registro de nuevas guardias.

Interacción con el Sistema de Liquidación de Haberes Provincial (SILIQIP): es necesario que SiWebGM se relacione indirectamente con el Sistema de Liquidación de Haberes Provincial, al momento de registrar una guardia, consultando las horas que debe cumplir el empleado en el

día, en el caso que posea un cargo de planta permanente en la Administración Pública. Se plantea que estos datos sean actualizados automáticamente desde el SILIQUIP en la B.D. de SiWebGM, cada vez que se realiza una liquidación desde dicho sistema.

4.2.2.2. Funciones de la Aplicación

El prototipo de Sistema Web para la Gestión de Guardias Médicas abarcará las siguientes funcionalidades:

- Gestión de los registros de guardias médicas realizadas por el personal en cada Área
- Gestión de los Servicios Internos que ofrece cada Área o Establecimiento Sanitario
- Gestión de días no laborables (asuetos o feriados)
- Gestión de Periodos de Liquidación de Área
- Emisión de reportes varios
- Gestión de usuarios del sistema

4.2.2.3. Características de los Usuarios

Para todos los niveles de usuario, se solicitará un nivel medio de estudios y experiencia básica en el manejo de sistemas informáticos en un ambiente Windows o Linux, como así también conocimientos mínimos en la utilización de navegadores web.

4.2.2.4. Restricciones

Restricciones de Interfaces de Aplicaciones: como se mencionó anteriormente, SiWebGM debe interactuar con otros sistemas software existentes, razón por la cual tiene que limitarse a utilizar las Bases de Datos ya existentes de dichos sistemas y a lo sumo agregar nuevas

estructuras de almacenamiento que no generen un alto impacto en la funcionalidad de los demás sistemas.

Restricciones de hardware: se requiere un equipo hardware servidor de potentes características para alojar el servidor Web que implementará la funcionalidad de la Aplicación Web. Los usuarios de la WebApp deben poseer una conexión a Internet para poder acceder a la misma.

Restricciones de lenguaje: las Bases de Datos de los sistemas software existentes están administradas por el Motor de Bases de Datos PostgreSQL, por lo que se deberá continuar utilizando este Administrador de B.D.

4.2.2.5. Suposiciones y Dependencias

En lo que se refiere puntualmente al Servidor Web donde se alojará el prototipo de Aplicación Web desarrollado, deberá contar con memoria principal de gran capacidad y de alta velocidad. También será importante que el procesador pueda efectuar un procesamiento rápido de todas las solicitudes, por lo que se recomienda que posea mínimamente doble núcleo.

En virtud de que el acceso al Servidor Web se realizará a través de una conexión a Internet, se debe tener en cuenta su configuración para permitir el mencionado acceso, sin olvidar las políticas de seguridad de la infraestructura de la red. Es de gran importancia verificar que la velocidad de transferencia de datos a través de dispositivos sea generosa, porque ella determinará la cantidad de conexiones posibles que pueda llegar a manejar el Servidor. Además se debe prever el suministro ininterrumpido de energía.

En lo que respecta a las computadoras que actuarán de clientes, simplemente deberán contar con un navegador web actualizado y que cumpla con los estándares internacionales y acceso a Internet.

4.2.2.6. *Requerimientos Diferidos*

Para el desarrollo de futuras versiones de la Aplicación Web, se prevé la implementación de un módulo para la gestión de licencias de las que pueden gozar los empleados que posean contratos de guardias.

Así también se planea implementar la gestión de auditorías por parte del Sistema, con el fin de registrar cada acción que realicen los distintos usuarios en la WebApp.

4.2.3. *Requerimientos Específicos*

4.2.3.1. *Requerimientos Funcionales*

A continuación se especifican los requisitos funcionales identificados para SiWebGM, los cuales definen las capacidades y servicios que el sistema software debe cumplir:

1. El sistema debe mantener y restringir accesos al sistema como se especificará más adelante, de los siguientes roles de usuario: Supervisor y Administrador de Guardias Médicas (AGM), donde esta última categoría contiene dos tipos de usuario que comparten la mayoría de la funcionalidad: AGM de SAF y AGM de Área.
2. El AGM de Área debe estar relacionado a un *Área* en particular y solo debe tener acceso general a la información referida a su Área (empleados, guardias, etc.).
3. El AGM de SAF debe estar relacionado a un *SAF* en particular y solo debe tener acceso general a la información referida a todas las Áreas pertenecientes a su SAF (empleados, guardias, etc.).

4. El AGM debe poder agregar, buscar, modificar o eliminar los Servicios Internos por Área.
5. El AGM debe poder registrar o eliminar un nuevo Periodo de Liquidación, por cada Área y periodo mensual. No se permite modificar datos del Periodo de Liquidación de Área. Dicho Periodo de Liquidación de Área, es el que se tomará para la liquidación propiamente dicha de las guardias realizadas en el Área.
6. El AGM debe poder agregar, buscar, modificar o eliminar las guardias médicas correspondientes a un Periodo de Liquidación de Área en particular, y a un contrato de guardia de empleado que esté vigente en el Área al momento de realizar una guardia.
7. El AGM debe poder generar un reporte, para un Periodo de Liquidación y Área en particular, donde se indique en detalle por cada empleado, las guardias realizadas por día, el tipo de las mismas (Profesional o Técnica y Activa o Pasiva) y la duración en horas.
8. El AGM debe poder generar un reporte, para un Periodo de Liquidación y Área en particular, donde se muestre un resumen para cada Servicio Interno del Área, de los totales de Guardias Activas o Pasivas y de acuerdo a la duración de las mismas.
9. El AGM debe poder generar un reporte, para un Periodo de Liquidación de Área y un agente en particular, donde se muestre el detalle de las guardias registradas en el mes, mostrando al final del reporte un resumen de totales de guardias.
10. El Supervisor debe poder realizar la gestión de usuarios del sistema: agregar, buscar, modificar y eliminar usuarios. El registro de un nuevo usuario de la WebApp debe ser solicitado por Nota dirigida al Organismo correspondiente.

11. El Supervisor debe poder agregar, buscar, modificar o eliminar días no laborables (asuetos o feriados) del año.
12. El Supervisor debe poder agregar, buscar, o eliminar registros de valores de duración de guardias preestablecidos por las normativas vigentes (por ejemplo: guardias de 8 horas, 12 horas, 16 horas, etc.)
13. El Supervisor debe poder realizar todas las acciones que realiza el AGM, para todas las Áreas y todos los SAF.

4.2.3.2. Requisitos que representan Reglas de Negocio.

Se detallan aquí las restricciones del dominio del problema que se deben tomar en cuenta junto con los requisitos funcionales.

(RN1) Una guardia médica consiste en: id periodo de Liquidación de Área, id contrato de guardia, id servicio interno de área, duración de la guardia, fecha, hora inicio guardia, tipo permanencia de guardia (activa o pasiva), tipo de guardia según título de empleado (Técnica o Profesional).

(RN2) La fecha de realización de una guardia debe coincidir con el mes y año del Periodo de Liquidación de Área.

(RN3) Un Empleado no puede realizar más de una guardia del mismo tipo (Activa o Pasiva), para una misma fecha y Área del Contrato.

(RN4) Un Empleado puede realizar guardias en una misma fecha y en varias Áreas, siempre y cuando la suma total de horas no supere las 24 horas, teniendo en cuenta: las horas que el empleado debe cumplir en el día por un cargo en Administración Pública Provincial (si lo tuviere y teniendo en cuenta si el día es Feriado), más las horas de las guardias ya registradas para el empleado, más las horas de la guardia a registrar.

(RN5) Los datos de un Periodo de Liquidación de Área consisten en: id Periodo de Liquidación, periodo mensual (ejemplo: 201506), Id Área,

estado (Cerrado o Abierto), fecha creación, fecha último proceso liquidación (éste último dato es completado por el SILIQGM cada vez que ejecute el proceso de liquidación de guardias para el Periodo correspondiente).

(RN6) Los datos de un registro de Día no Laborable consisten en: fecha y motivo.

(RN7) Los datos de un Servicio Interno de Área consisten en: ID Área a la que pertenece el Servicio Interno, nombre y el campo activo (SI/NO).

(RN8) No se permite agregar, modificar o eliminar guardias cuando el Periodo de Liquidación de Área a la que corresponden, se encuentre con estado "Cerrado" por haber concluido proceso de liquidación de guardias.

(RN9) Solo se permiten modificar los siguientes campos de una guardia ya registrada: Servicio Interno del Área del Contrato, duración, fecha, hora y tipo permanencia de guardia (activa o pasiva).

(RN10) No se podrán eliminar registros de: Servicios Internos de Área, Periodos de Liquidación de Área y Días no laborables, cuando existan guardias relacionadas a estos registros.

(RN11) El AGM de Área sólo debe poder recuperar registros de Empleados que posean un contrato de guardia vigente en el Área a la que pertenece el usuario.

(RN12) El AGM de SAF sólo debe poder recuperar registros de Empleados que posean contratos de guardias vigentes en las Áreas del SAF al que pertenece el usuario.

(RN13) El Supervisor debe poder recuperar registros de Empleados que posean contratos de guardias vigentes en todas las Áreas y todos los SAF.

(RN14) Los datos de un Usuario (AGM y Supervisor) consisten en: Nombre Usuario, N° CUIL, Apellido y Nombre, e-mail, teléfono de

contacto, Área o SAF relacionada, tipo de usuario (AGM de Área / AGM de SAF / Supervisor), activo (SI / NO), fecha alta usuario.

4.2.3.3. Requerimientos No Funcionales

1. El prototipo de Aplicación Web propuesto debe cumplir con el Principio “Comprensible” Nivel A de las **Pautas de Accesibilidad al Contenido Web 2.0**, con el objetivo de proporcionar información y manejo de interfaces de usuario *comprensibles*.
2. La presentación visual de la aplicación web debe respetar los estándares del W3C (World Wide Web Consortium) en cuanto al diseño de las páginas web HTML, CSS (Hojas de Estilos) se refiere.
3. El prototipo de Aplicación Web debe ofrecer un plan de seguridad que contemple: identificación, autenticación, control de acceso y autorización para los diferentes tipos de usuarios, cada uno con distintos niveles de acceso y operabilidad en la Aplicación.

4.2.4. Glosario

Contrato de guardia: Relación Laboral de Contrato que se realiza para habilitar a una persona para realizar guardias médicas en un establecimiento asistencial (Área) determinado e indicando el tipo de guardia autorizado a realizar: profesional o técnica, según la capacitación que posea la persona al momento de firmar el contrato.

Guardia profesional: es aquella guardia médica que es realizada por un empleado de guardia y que posee un contrato autorizando a realizar guardias profesionales.

Guardia técnica: se refiere a una guardia médica que es realizada por un empleado de guardia y que posee un contrato autorizando a realizar guardias técnicas.

Guardia Activa: guardia medica que requiere la permanencia del personal sanitario en el lugar de trabajo en el horario estipulado, y desarrollando las funciones encomendadas. Debe estar estipulada en el Contrato de Guardia médica.

Guardia Pasiva: guardia médica que se presenta cuando el personal no tenga presencia constante en el lugar de trabajo pero permanezca en disponibilidad y concurra al lugar que se le ha destinado dicha guardia pasiva de forma inmediata y cuando le sea requerido por los medios preestablecidos. Debe estar estipulada en el Contrato de Guardia médica.

Servicio Interno de Área: división interna con la que cuenta un Área, y que corresponde generalmente a las especialidades médicas cubiertas por las guardias en dicha Institución.

4.3. MODELADO DEL PROTOTIPO DE APLICACIÓN WEB

A continuación, se exponen los modelos resultantes luego de haber aplicado el enfoque UWE (UML-based web engineering) al modelado y diseño de la WebApp.

4.3.1. Modelo de Requerimientos (Casos de Uso)

4.3.1.1. Actores del Sistema

Los Actores principales del Prototipo de Aplicación Web propuesto son:

Administrador de Guardias Médicas de SAF (AGM de SAF): este usuario está asociado a un SAF en particular, y tiene la responsabilidad de la gestión de los registros de guardias médicas correspondientes a los empleados que posean contratos de guardias en Áreas que pertenezcan al SAF específicamente.

Administrador de Guardias Médicas de Área (AGM de Área): este usuario está asociado a una Área en particular, y tiene la responsabilidad

de la gestión de los registros de guardias médicas correspondientes a los empleados que posean contratos de guardias en esa Área específicamente.

Supervisor: usuario que tiene la responsabilidad de gestionar los parámetros del sistema, como la gestión usuarios del sistema, días no laborables, etc. y además tiene acceso a la misma funcionalidad que los usuarios Administrador de Guardias Médicas de SAF y de Área.

En la Figura 28 se muestra el diagrama de actores de la aplicación web propuesta, donde se modela el actor “Administrador de Guardias Médicas” como una generalización de los actores AGM de SAF, AGM de Área y Supervisor, y luego el actor “Usuario” es una generalización del actor Administrador de Guardias Médicas, ya que existen funcionalidades comunes a todos los usuarios de la aplicación.

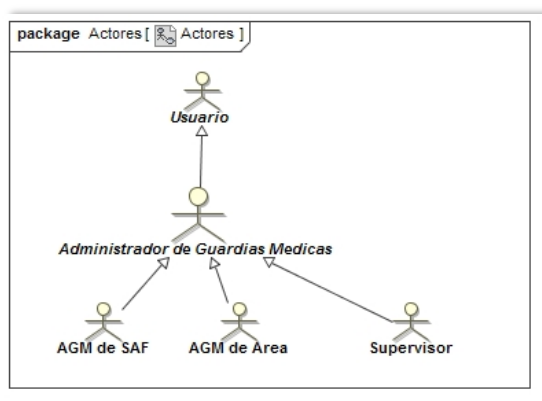


Figura 28: Actores del Prototipo SiWebGM

4.3.1.2. Diagramas de Casos de Uso

Para una mejor organización, los casos de uso se estructuraron en paquetes, de acuerdo a la funcionalidad por cada tipo de actor. Los paquetes resultantes son: *Acceso al Sistema*, *Administrar Guardias Médicas* y *Administrar Parámetros*.

↪ Paquete: Acceso al Sistema

En la Figura 29 se puede visualizar el paquete “Acceso al Sistema”, que expone la funcionalidad típica de un usuario de este tipo de aplicaciones.

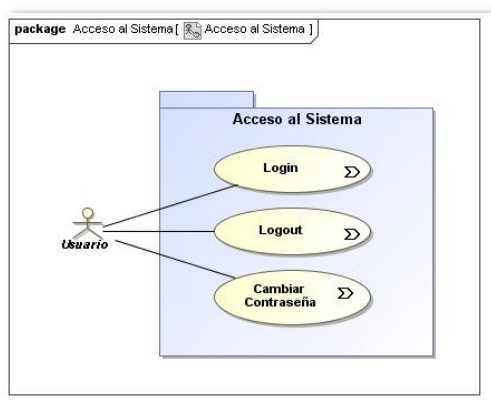


Figura 29: Diagrama de Casos de Uso para el paquete “Acceso al Sistema”

↪ Paquete: Administrar Guardias Médicas

En la Figura 30 se observa el paquete denominado “Administrar Guardias Médicas”, que representa la funcionalidad central del prototipo de Aplicación Web propuesto.

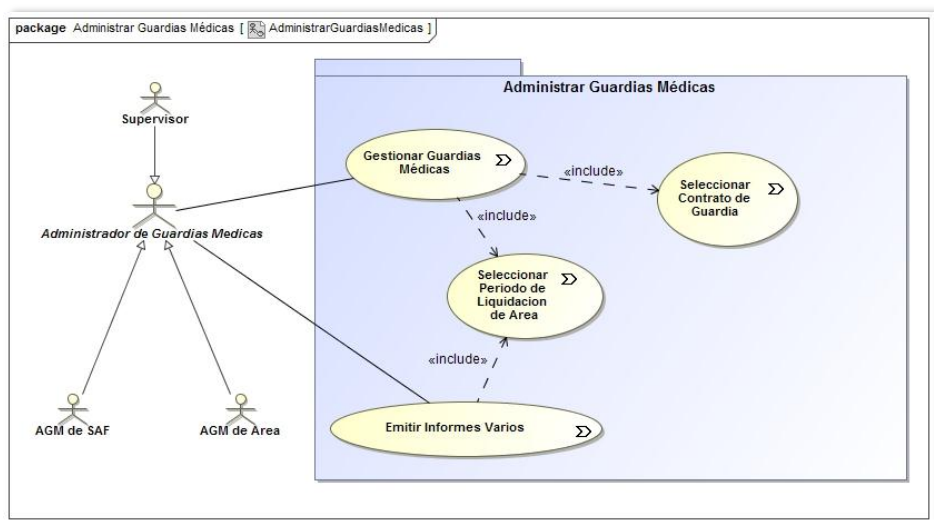


Figura 30: Diagrama de Casos de Uso para el paquete “Administrar Guardias Médicas”

A modo general, el caso de uso “Gestionar Guardias Médicas” engloba la funcionalidad de agregar, modificar o eliminar registros de guardias médicas. Según lo especificado, una guardia médica debe estar asociada a un Periodo de Liquidación de Área y un Contrato de Guardia en particular, es por ello que se decidió separar en dos casos de uso diferentes la selección de estas entidades, para que puedan ser reutilizados por otros casos de uso.

📁 Paquete: Administrar Parámetros

Por último, se muestra a continuación en la Figura 31, los casos de uso del paquete “Administración de Parámetros”:

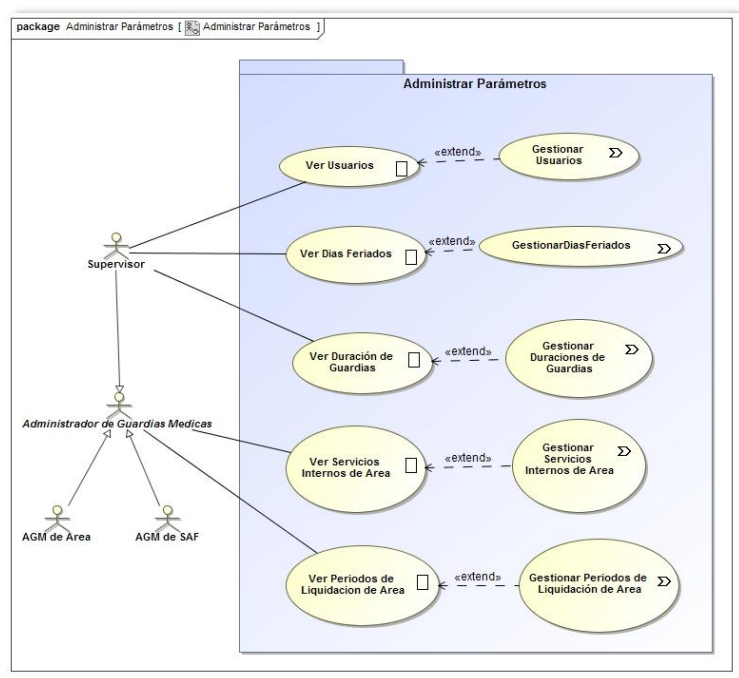


Figura 31: Diagrama de Casos de Uso para el paquete “Administración de Parámetros”

Como se aprecia en la figura anterior, la gestión de una entidad o parámetro en particular, consta de un caso de uso que envuelve la funcionalidad de mostrar los registros correspondientes y luego dar al usuario la opción de agregar, modificar o eliminar los mismos mediante los casos de uso “Gestionar”.

Así también, se observa que el Actor Supervisor es el único que tiene acceso a los casos de uso referidos a la gestión de usuarios, feriados y duración de guardias. Además, debido a la relación de generalización con el actor AGM, el supervisor también tiene acceso a la misma funcionalidad que el mencionado AGM.

4.3.1.3. Descripción textual de Casos de Uso

A continuación, en la Tabla 6, Tabla 7, Tabla 8 y Tabla 9 respectivamente se consignan descripciones textuales que complementan la especificación de los casos de uso del paquete “Administrar Guardias Médicas”, los cuales representan la principal funcionalidad de la WebApp propuesta. La descripción de los demás Casos de Uso se puede consultar en la sección Anexo B.

Tabla 6: Descripción del Caso de Uso: Gestionar Guardias Médicas

Iniciador	AGM, Supervisor
Precondiciones	El actor debe haber ejecutado satisfactoriamente el Caso de Uso “Iniciar Sesión”
Flujo básico	
1. Actor: elige la opción de Menú “Gestionar Guardias Médicas”	
2. Sistema: se incluye Caso de Uso “Seleccionar Periodo de Liquidación de Área”	
3. Sistema: teniendo en cuenta el Periodo de Liquidación de Área seleccionado en el punto anterior, se incluye Caso de Uso “Seleccionar Contrato de Guardia”	
4. Sistema: de acuerdo al Periodo de Liquidación de Área y Contrato de Guardia seleccionado por el Actor en los CU incluidos anteriormente, el sistema realiza una búsqueda en la B.D. y presenta una página web con una lista de las guardias médicas recuperadas.	
5. Sistema: muestra para cada registro de la lista de guardias recuperada: Nombre Servicio Interno de Área, Duración de Guardia, Fecha, Hora, Tipo Permanencia Guardia y botones con la opción de “Modificar” o “Eliminar” cada guardia	
6. Sistema: se muestra además, al final de la lista de guardias, la opción de “Agregar Guardia Medica” o volver al Inicio de la WebApp.	
7. Sistema: muestra la ruta de navegación referida a la Gestión de Guardias Médicas, brindando la opción de volver a “Seleccionar Periodo de Liq. Área” o “Seleccionar Contrato de Guardia”	
8. Actor: escoge una de las opciones presentadas, tantas veces como desee: <ul style="list-style-type: none"> a. Agregar una guardia (continúa en Sección A) 	

b. Modificar una guardia (continúa en Sección B) c. Eliminar una guardia (continúa en Sección C)	
9. Actor: escoge la opción “Volver a Inicio de WebApp”	
10. Sistema: el CU finaliza	
Poscondiciones	Se agregan, modifican o eliminan guardias realizadas en un Periodo de Liquidación de Área correspondientes a un Contrato de Guardia de Empleado, en el Área relacionada con el Periodo de Liquidación
SECCION A – Agregar una Guardia	
Flujo básico	
1. Sistema: muestra una página web con los datos del Periodo de Liq. Area, Contrato de Empleado seleccionados anteriormente y un formulario con los campos a completar para la guardia a registrar: Servicio Interno de Área, duración de la guardia, fecha, hora inicio y tipo permanencia guardia.	
2. Sistema: para el campo <i>Servicio Interno de Área</i> , el sistema construye y muestra un campo de entrada tipo selección que contiene una lista de los Nombres de Servicios Internos de Área <i>habilitados</i> de acuerdo al Área del Periodo de Liquidación seleccionado.	
3. Sistema: también para los campos <i>duración de guardia</i> y <i>tipo permanencia guardia</i> , el sistema construye y muestra un campo de entrada tipo selección que contienen una lista con las opciones válidas para cada uno.	
4. Actor: selecciona el Servicio Interno donde se realizó la guardia de la lista de selección.	
5. Actor: selecciona la duración de la guardia de las opciones disponibles de la lista de selección.	
6. Actor: ingresa fecha y hora inicio de la guardia.	
7. Actor: selecciona el tipo permanencia guardia de las opciones disponibles de la lista de selección.	
8. Actor: indica al Sistema que desea guardar los datos de la guardia.	
9. Sistema: verifica que los datos de la guardia a registrar cumplan la regla de negocio RN1, RN2, RN3 y RN8.	
10. Sistema: recupera la información necesaria para realizar verificaciones según la regla de negocio RN4.	
11. Sistema: registra la nueva guardia e informa al AGM.	
12. Sistema: muestra una página web con la <i>lista de guardias</i> actualizada para el Contrato y Periodo de Liquidación de Área.	
13. Sistema: finaliza la sección A.	
Flujo alternativo 1	Punto 8. El AGM decide cancelar la registración de la guardia. El sistema vuelve a mostrar la página con la lista de guardias para el Contrato y Periodo de Liquidación de Área.
Flujo alternativo 2	Punto 9 y 10. No se cumplen con las Reglas de Negocio especificadas en estos puntos. El sistema muestra mensaje de

	alerta, no se registra la guardia y solicita se revisen los datos ingresados.
SECCION B – Modificar Guardia	
Flujo básico	
1. Sistema: muestra una página web con un formulario que contiene los datos de la guardia seleccionada. Se tiene en cuenta regla de negocio RN9.	
2. Actor: realiza todas las modificaciones que desea sobre los campos habilitados.	
3. Actor: indica al Sistema que desea registrar los cambios realizados a la guardia.	
4. Sistema: verifica que los datos ingresados cumplan con las regla de negocio RN1, RN2, RN3, RN4 y RN8.	
5. Sistema: registra los cambios realizados a la guardia e informa al AGM.	
6. Sistema: muestra una página web con la <i>lista de guardias</i> actualizada para el Contrato y Periodo de Liquidación de Área.	
7. Sistema: finaliza la sección B.	
Flujo alternativo 1	Punto 3. El Actor decide cancelar la modificación de la guardia. El sistema vuelve a mostrar la página web con la lista de guardias para el Contrato y Periodo de Liquidación de Área.
Flujo alternativo 2	Punto 4. No se cumplen con las reglas de negocio especificadas. El sistema muestra mensaje de alerta y solicita se revisen los datos ingresados.
SECCION C – Eliminar Guardia	
Flujo básico	
1. Sistema: pide al Actor una confirmación para eliminar la guardia seleccionada.	
2. Actor: confirma la eliminación de la guardia.	
3. Sistema: realiza verificación de Periodo Liquidación de Área según RN8.	
4. Sistema: elimina la guardia seleccionada.	
5. Sistema: finaliza la sección C.	
Flujo alternativo 1	Punto 2. El usuario decide cancelar la eliminación de la guardia. El sistema no elimina la guardia.
Flujo alternativo 2	Punto 3. No se cumple la RN8, el sistema informa al usuario y no elimina la guardia.

Tabla 7: Descripción del Caso de Uso: Seleccionar Periodo de Liquidación de Área

Iniciador	Caso de Uso “ <i>Gestionar Guardias Médicas</i> ” o Caso de Uso “ <i>Emitir Informes Varios</i> ”
Otros actores	AGM, Supervisor

Flujo básico	
1. Sistema: presenta una lista de Periodos de Liquidación para las Áreas a las que tenga acceso el actor, de acuerdo a RN11, RN12 y RN13. También se tiene en cuenta la RN8, por lo tanto solo se muestran los registros de Periodos con estado igual a "Abierto", en el caso de que el iniciador sea el CU Gestionar Guardias Médicas.	
2. Sistema: los datos que muestra el sistema para la Lista de Periodos son: Nro. Liquidación, Periodo mensual, Nombre Área, Estado	
3. Sistema: muestra el dato "Nro. Liquidación" como un link, mediante el cual se devuelve el ID del Periodo de Liquidación de Área seleccionado al Caso de Uso Base.	
4. Actor: selecciona un Periodo de Liquidación de Área en particular, haciendo un clic en el link "Nro. Liquidación".	
5. Sistema: devuelve al CU base los datos del Periodo de Liquidación de Área seleccionado por el actor. El CU finaliza y continúa la ejecución en el Caso de Uso Base.	
Poscondiciones	Se devuelve al CU Base los datos de un Periodo de Liquidación de Área en particular.

Tabla 8: Descripción del Caso de Uso: Seleccionar Contrato de Guardia

Iniciador	Caso de Uso " <i>Gestionar Guardias Médicas</i> "
Otros actores	AGM, Supervisor
Flujo básico	
1. Sistema: presenta una página web con una lista de Contratos de Guardias de Empleados, que pertenecen al Área del Periodo de Liquidación de Área seleccionado en el CU Base.	
2. Sistema: los datos que muestra el sistema en la Lista de Contratos de Guardia son: CUIL, Nombre Empleado, Nombre Área del Contrato, Fecha Inicio, Fecha Fin de Contrato	
3. Sistema: muestra el dato "CUIL" como un link, mediante el cual se devuelve el ID del Contrato de Guardia seleccionado al Caso de Uso Base	
4. Actor: selecciona un Contrato de Guardia de Empleado en particular, haciendo un clic en el link "CUIL".	
5. Sistema: devuelve al CU base los datos del Contrato de Guardia seleccionado por el actor. El CU finaliza y continúa la ejecución en el Caso de Uso Base.	
Poscondiciones	Se devuelve al CU Base los datos de un Contrato de Guardia de Empleado en particular.

Tabla 9: Descripción del Caso de Uso: Emitir Informes Varios

Iniciador	AGM, Supervisor
Precondiciones	El actor debe haber ejecutado satisfactoriamente el Caso de Uso

	"Iniciar Sesión"
Flujo básico	
1. Sistema: presenta una página web con 3 opciones de reporte de guardias: detalle, resumen o guardias por agente.	
2. Sistema: se incluye CU "Seleccionar Periodo de Liquidación de Área"	
3. Actor: selecciona el tipo de reporte que desea generar	
4. Actor: indica que se genere el reporte en base a las selecciones realizadas.	
5. Sistema: confecciona reporte en base a los datos seleccionados y presenta al usuario el informe en formato "pdf". El CU finaliza.	
Flujo alternativo 1	Punto 3. El usuario elije el tipo de reporte a generar "guardias por agente", entonces el Sistema habilita un campo de texto para que el usuario ingrese el CUIL del agente a consultar.
Poscondiciones	Se genera un informe en formato "pdf" para un Periodo de Liquidación de Área en particular.

4.3.2. Modelo de Contenido

A continuación en la Figura 32 se muestra el *Modelo de Contenido* para el prototipo de Aplicación Web de Guardias Médicas, el cual fue diseñado en base al *Modelo de Requerimientos*. El propósito de este modelo es proveer una especificación visual de la información relevante del dominio que comprende principalmente el contenido propiamente dicho de la Aplicación Web.

realizar la navegación utilizando elementos de acceso tales como índices, visitas guiadas, consultas y menús.

En la Figura 33 se muestra un diagrama correspondiente al *Modelo de Navegación* resultante para el Rol del usuario Administrador de Guardias Médicas, el cual fue diseñado a partir del *análisis de requerimientos* y del *Modelo de Contenido*. En dicha figura se puede observar que se ha definido un patrón de navegación en común respecto a la gestión de entidades (guardias, servicios internos, etc.). En primer lugar, el sistema web presenta una página web con una lista de registros de la entidad a gestionar de acuerdo a los alcances del usuario, por ejemplo, para un usuario AGM de SAF, se mostrarán en la lista de Periodos de Liquidación de Área solo aquellos registros cuyas Áreas pertenezcan al SAF del usuario. Luego por cada registro, en la mayoría de los casos, se brinda la opción de modificar o eliminar el mismo. Así también, se muestra la opción de poder agregar un nuevo registro.

La navegación de la gestión de Guardias Médicas es un caso especial, ya que para poder mostrar una lista de las guardias realizadas, el usuario debe en primer lugar seleccionar un Periodo de Liquidación de Área, luego el sistema presenta una lista de los Contratos de Guardias correspondientes al Área del Periodo de Liquidación de Área seleccionado. Finalmente, en base al Contrato de Guardia y Periodo de Liquidación de Área elegidos por el usuario, el sistema web presenta una lista de las guardias realizadas.

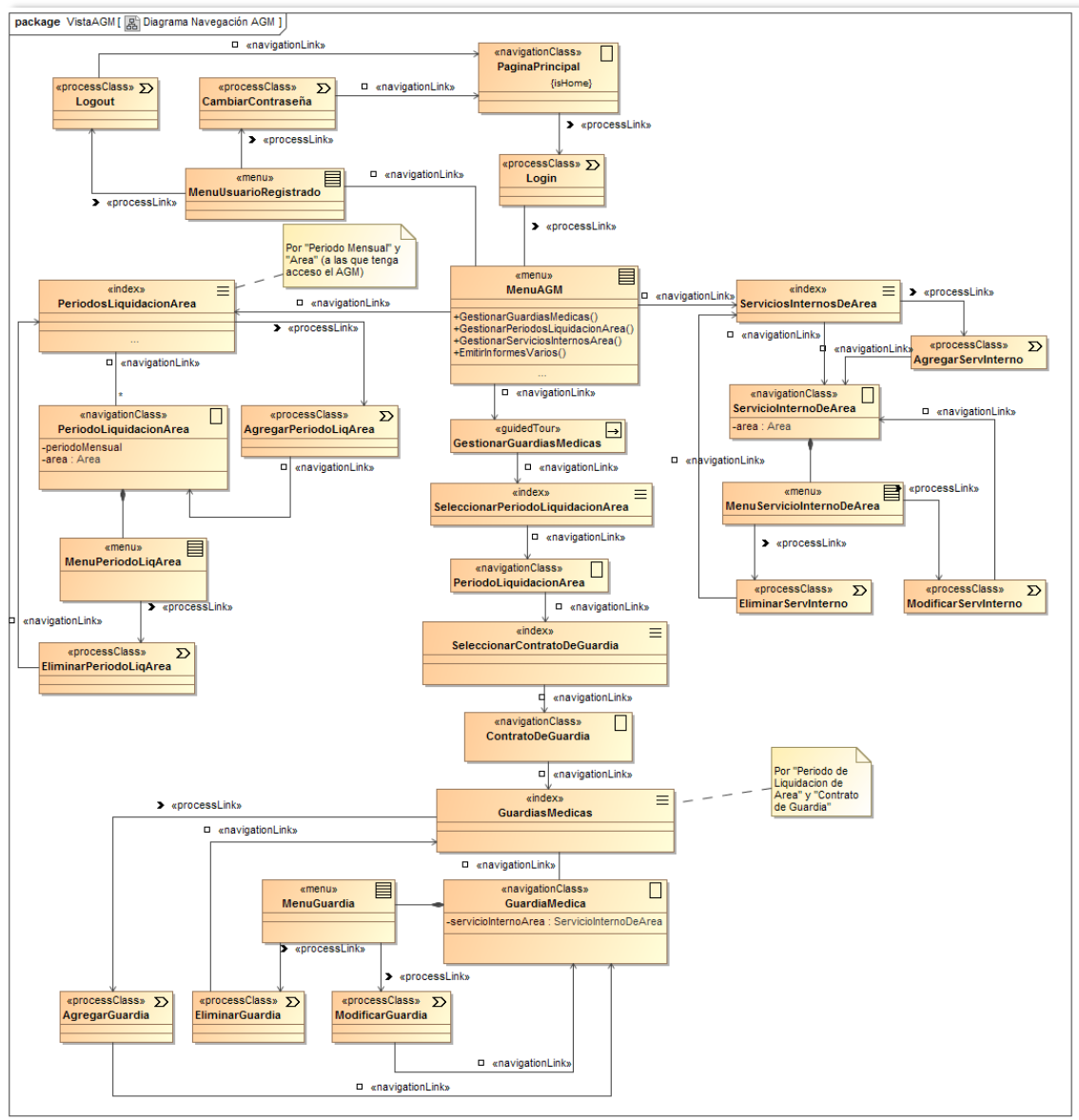


Figura 33: Diagrama de Navegación para el Rol AGM

La navegación para el Rol de Supervisor es igual a la del AGM, y además se agregan las opciones de Menú: “Gestión de Días Feriados”, “Gestión de Duración de Guardias” y “Gestión de Usuarios”.

4.3.4. Modelo de Presentación

El *Modelo de Presentación* se basa en el *Modelo de Navegación* y describe qué elementos (por ejemplo texto, elementos, links, formularios) se utilizarán para presentar los nodos de navegación.

Las propiedades de accesibilidad al contenido web que debe satisfacer el prototipo web desarrollado, están muy relacionadas a los aspectos del lenguaje de marcado HTML y a la presentación de las páginas web. Por ello, creemos oportuno durante el modelado de la presentación, hacer mención a las características de accesibilidad que se deben cumplir, en base a los atributos del árbol de requerimientos definidos en la Sección 3.2.2 del presente trabajo, de tal manera de que se incorporen dichas características desde el inicio del desarrollo de la WebApp.

Desde la Figura 34 hasta la Figura 39, se pueden observar los fragmentos de diagramas que componen el modelo final de presentación para el actor AGM, donde se representan las páginas web y componentes de la interfaz de usuario (UI) del prototipo Web, proporcionando una vista abstracta del mismo.

Así también, se indican en recuadros de color turquesa, en qué punto se deben incluir los requerimientos de accesibilidad web según el modelo de calidad desarrollado. Por ejemplo, en la Figura 34, se indica que para el elemento “PaginaPrincipal” del tipo “<presentationPage>”, se debe especificar el lenguaje humano de la página mediante el atributo “lang” del elemento <html>, de acuerdo a lo especificado en el Atributo 1.1.1 del árbol de requerimientos de calidad.

Para el modelado de la presentación, se utilizó el elemento <presentationPage>, para representar el esquema principal de las páginas web de la webapp, compuesto por un encabezado y pie de página (igual para todas las páginas web) y en la porción central del mismo, contiene el elemento <presentationAlternatives>, indicando con

esto que la webapp debe presentar el formulario de Login, o bien, el grupo de presentación “Contenido AGM”, dependiendo de que no haya un usuario identificado en el sistema, o que el usuario AGM haya ingresado a la webapp, respectivamente (Ver Figura 34)

Luego, en la Figura 35 se expande la especificación del componente “<presentationGroup> Contenido AGM”, indicando los elementos que lo componen como ser:

- ✓ un elemento del tipo <presentationGroup> que contiene una lista de enlaces que representan el menú de opciones para el actor AGM
- ✓ otro elemento del tipo <presentationGroup> que representa el menú de usuario conteniendo etiquetas de texto con los datos del mismo (nombre, rol, Área o SAF de acceso) y un botón para cerrar la sesión de usuario actual.
- ✓ un elemento <presentationAlternatives> que contiene distintas alternativas de clases de presentación con el nombre del contenido que se mostrará de acuerdo a la opción de menú que seleccione el usuario (MsjInicio, ContenidoGuardiasM, ContenidoPeriodoLiquidacion, ContenidoServicioInternoArea, ContenidoReportes). A su vez, cada uno de estos elementos, se expande para detallar su contenido, tal como se muestra en la Figura 36, Figura 37, Figura 38 y Figura 39 respectivamente, comenzando por “ContenidoGuardiasM”.

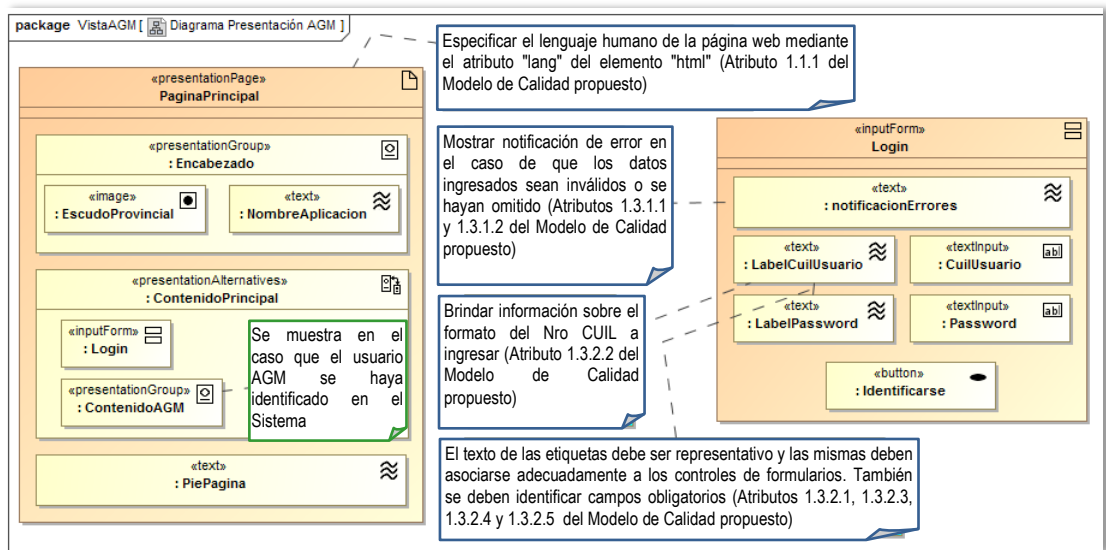


Figura 34: Fragmento de Diagrama de Presentación para la página principal de la WebApp y login de usuario

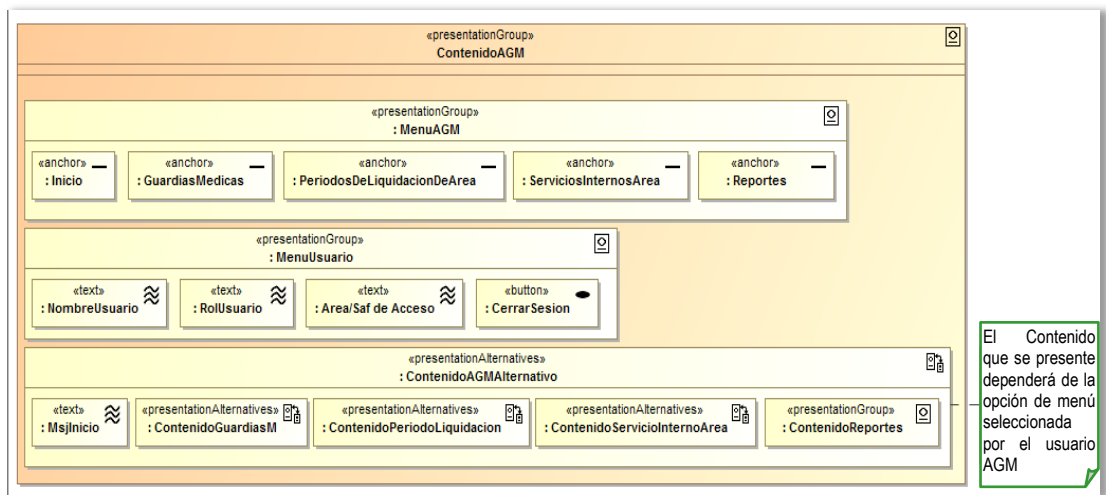


Figura 35: Fragmento de Diagrama de Presentación de menú principal y menú de usuario para el actor AGM

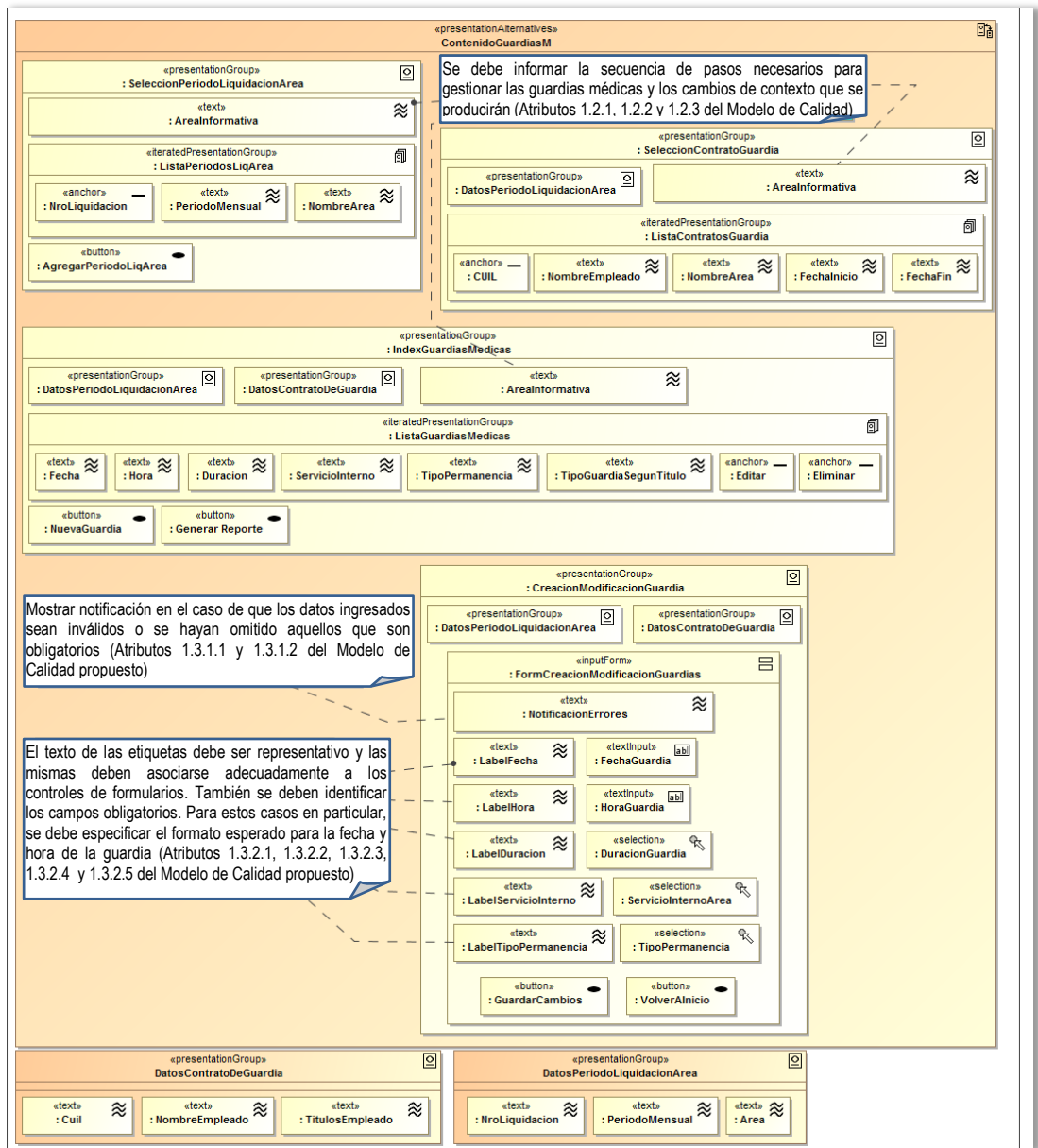


Figura 36: Fragmento de Diagrama de Presentación de componentes de IU involucrados en la Gestión de Guardias Médicas

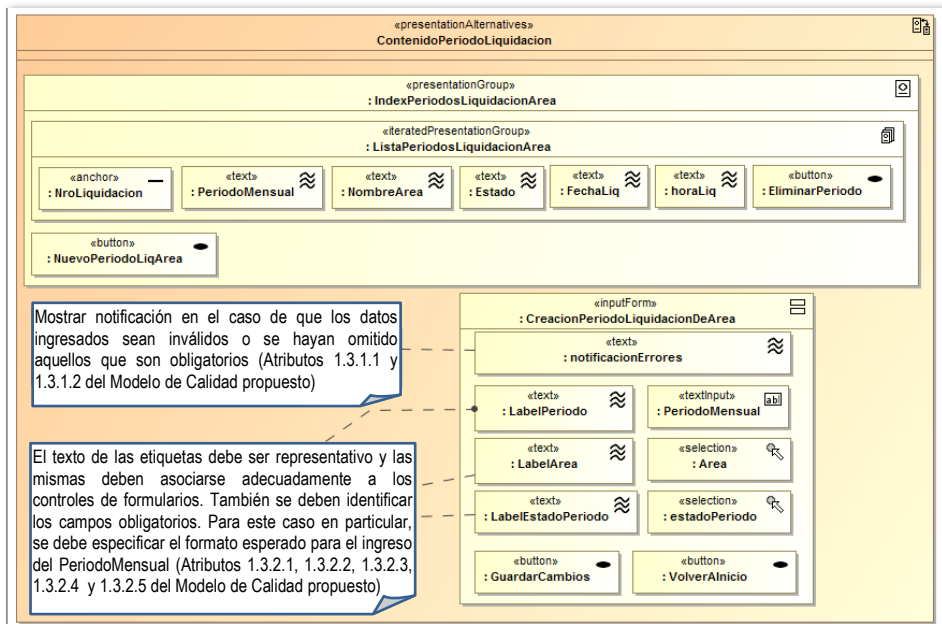


Figura 37: Fragmento de Diagrama de Presentación de componentes de IU involucrados en la Gestión de Periodos de Liquidación de Área

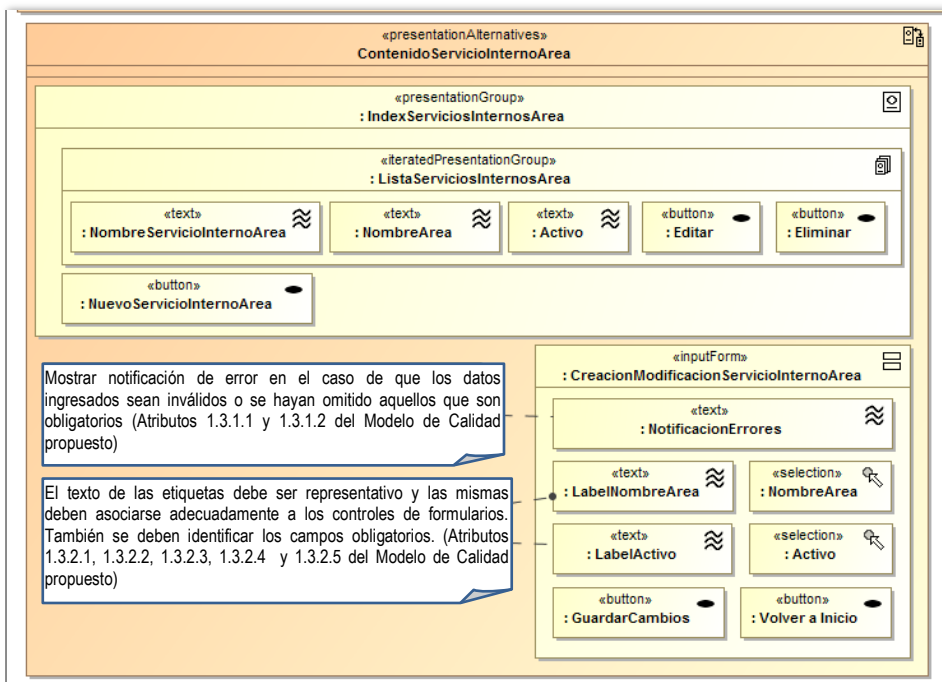


Figura 38: Fragmento de Diagrama de Presentación de componentes de IU involucrados en la Gestión de Servicios Internos de Área

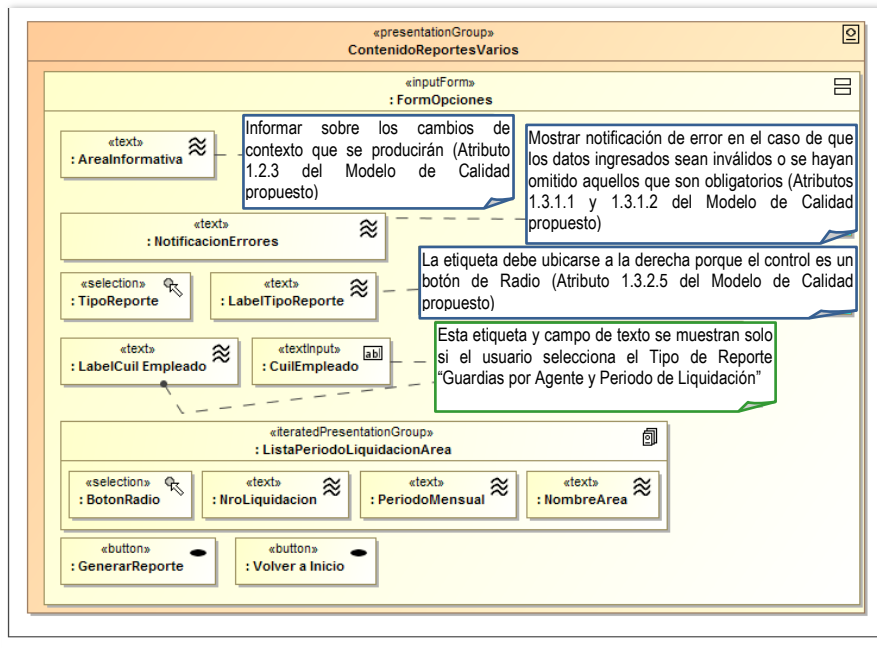


Figura 39: Fragmento de Diagrama de Presentación de componentes de IU para la página web de emisión de reportes varios

CAPITULO V – PRESENTACION DEL PROTOTIPO SIWEBGM

5.1. INTRODUCCION

El objetivo de este capítulo es presentar el prototipo “Sistema Web de Gestión de Guardias Médicas (SiWebGM)” desarrollado.

Para ello, se confeccionará esta sección dándole el enfoque de un “Manual de Usuario”, considerando al lector como un posible usuario de la WebApp, brindando de esta manera, una ayuda que le permita al mismo operar la aplicación correctamente, acercándolo a las distintas funcionalidades de la citada herramienta.

Esta sección estará sujeta a actualizaciones y modificaciones periódicas por parte del equipo encargado del desarrollo y mantenimiento del sistema, ante eventuales cambios, actualizaciones o futuros desarrollos de la aplicación.

5.1.1. Alcance

La presente sección describe la operación funcional del sistema web y está destinado a todos los Roles de usuarios definidos en la Aplicación: Supervisor, Administrador de Guardias Médicas de SAF y Administrador de Guardias Médicas de Área.

5.1.2. Siglas y Abreviaturas

SiWebGM: Sistema Web de Gestión de Guardias Médicas

SAF: Servicio Administrativo Financiero

AGM de SAF: Administrador de Guardias Médicas de SAF (Rol Usuario)

AGM de Área: Administrador de Guardias Médicas de Área (Rol Usuario)

5.2. ESTRUCTURA GENERAL DE LA WEBAPP

5.2.1. Esquema General de la Aplicación

La WebApp tiene un esquema gráfico que en general se presenta en todas las páginas web que la componen. A continuación, en la Figura 40 se detallan las áreas más comunes:




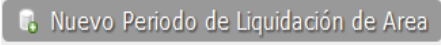


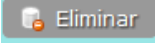
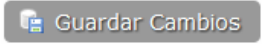
Figura 40: Esquema general de la WebApp

5.2.2. Imágenes y Botones gráficos

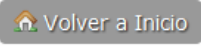
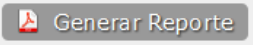


A lo largo de la aplicación, se presentarán situaciones donde el usuario deba decidir sobre una acción a seguir con respecto a diversas alternativas. En estos casos, la WebApp presentará la posibilidad de utilizar botones. Para identificarlos, el usuario debe posicionarse con el "mouse" sobre un botón y en forma inmediata, se visualizará el texto que indica su utilidad.

Así también, se mostrarán párrafos o listas con un icono específico que servirán de información para el usuario.

A continuación, se expone la representación gráfica de cada uno, y su funcionalidad.

-  ➔ Estos botones se ubican en la sección superior de cada página de “Gestión de registros”, e indican la ruta de navegación actual, indicando la secuencia de pasos correspondientes a dicha gestión de entidades. Haciendo un clic en cada botón, la WebApp muestra una página según la opción seleccionada, ya sea “Lista de registros”, “Edición de Registro” o “Nuevo Registro”.
-  ➔ El texto en general de este botón es “Nuevo <NombreEntidad>”, y a través de esta opción, el sistema habilita una página con un formulario web para ingresar datos de la entidad correspondiente. En este caso, la entidad es “Periodo de Liquidación de Área”
-  ➔ Estos botones de paginación se presentan debajo de una lista de registros, para poder visualizar más resultados, si existieran, en la siguiente página (botón Siguiente), o en una página en particular, haciendo clic en un número específico.
-  ➔ Este botón se muestra por cada registro de una lista de entidades. Presionando el mismo, el sistema muestra una página web que contiene un formulario con los datos del registro seleccionado, para que el usuario ingrese las modificaciones correspondientes.
-  ➔ Al igual que el caso anterior, el botón “Eliminar” se muestra por cada registro de una lista de entidades. Presionando el mismo, se elimina el registro correspondiente.
-  ➔ Este botón se muestra en las páginas habilitadas para modificar o agregar un nuevo registro de una entidad.

Presionando el mismo le permite almacenar la información dentro del sistema.

-  ➔ Al igual que en el caso anterior, este botón se muestra en las páginas de modificación o alta de nuevos registros. En este caso, si el usuario presiona el mismo, NO se guardan los datos ingresados en el formulario web y la WebApp se dirige a la página de inicio de la misma.
-  ➔ Mediante este botón, se genera un informe en formato “pdf” de los datos provenientes de una consulta o lista.
-  ➔ Este icono indica una sección de comentarios informativos, sobre la página web actual.
-  ➔ Este icono indica una sección de advertencias, donde el sistema informa sobre inconsistencias detectadas en el ingreso de datos.

5.3. INGRESO A LA APLICACIÓN WEB

Para ingresar a la aplicación web deberá ingresar a la URL: <http://181.15.95.194/gm> desde un navegador web, como ser: Firefox²⁴, Google Chrome²⁵, entre otros. En la Figura 41 se muestra la página web principal de la aplicación, donde deberá ingresar los datos de usuario asignados oportunamente, de acuerdo al Rol correspondiente:

NOTA: se ha generado un usuario ficticio con Rol de Supervisor de modo tal que el lector pueda visitar la Aplicación Web y comprobar su funcionalidad. Los datos del mismo son: CUIL = 27273511720, Contraseña = 902ERT

²⁴ Sitio Web de Firefox: <https://www.mozilla.org> - Mozilla Foundation (US)

²⁵ Sitio web de Google Chrome: <https://www.google.com/chrome/browser/desktop/index.html>



Figura 41: Página principal y formulario de identificación de usuario

5.4. OPCIONES DEL SISTEMA SEGÚN ROLES DE USUARIOS

Como se mencionó en la sección anterior, SiWebGM contempla 3 tipos de Roles de Usuario: Supervisor, AGM de SAF y AGM de Área.

El usuario AGM de Área sólo tendrá acceso a la gestión de guardias y toda la información asociada, de una determinada Área en particular. De manera similar, el usuario AGM de SAF podrá gestionar las guardias asociadas a todas la Áreas de un SAF específico y por último, el usuario Supervisor tendrá acceso a los datos de guardias pertenecientes a todos los SAF, es decir, el universo completo. Además, este último rol de usuario, contará con otras opciones de administración de parámetros a las cuales no tienen acceso los roles AGM de SAF y AGM de Área.

5.4.1. Opciones de Menú en común para los tres Roles

En esta sección se expondrán las opciones de menú de los roles AGM de SAF y AGM de Área, ya que son las mismas para ambos y a su vez, el rol Supervisor también tendrá acceso a dichas opciones.

Entonces, una vez que el usuario con alguno de los roles AGM de SAF o AGM de Área se identifica en el sistema, se muestra la siguiente página web y menú principal situado en la parte superior de la misma (Figura 42):



Figura 42: Opciones de Menú para los usuarios AGM de SAF y de Área

Así también, debajo del menú principal, se puede observar una barra que contiene los siguientes datos del usuario identificado en el sistema: *Nombre usuario, Rol, Área o SAF de acceso*. También se presenta un botón para “Cerrar Sesión” actual.

5.4.1.1. Opción de Menú: “Guardias Médicas”

A través de esta opción se pueden gestionar los registros de las guardias médicas realizadas, es decir, se permiten agregar, modificar o eliminar guardias médicas.

Para la gestión de las guardias propiamente dichas, es necesario seguir una secuencia de pasos:

1. Seleccionar un “Periodo de Liquidación de Área”.
2. Seleccionar un “Contrato de Guardia” asociado al Área del “Periodo de Liquidación de Área” escogido en el paso anterior.

3. Agregar, modificar o eliminar registros de guardias médicas correspondientes al “Contrato de Guardia” y “Periodo de Liquidación de Área” seleccionados anteriormente.

A continuación, en la Figura 43 se muestra la primera fase de la gestión de guardias médicas, donde se muestra una lista de Periodos de Liquidación de Área, según las Áreas a las que tenga acceso el usuario identificado y que a la vez, dichos registros de Periodos se encuentren en estado “Abierto”, es decir, que aún no haya concluido para ellos el proceso de liquidación de guardias:

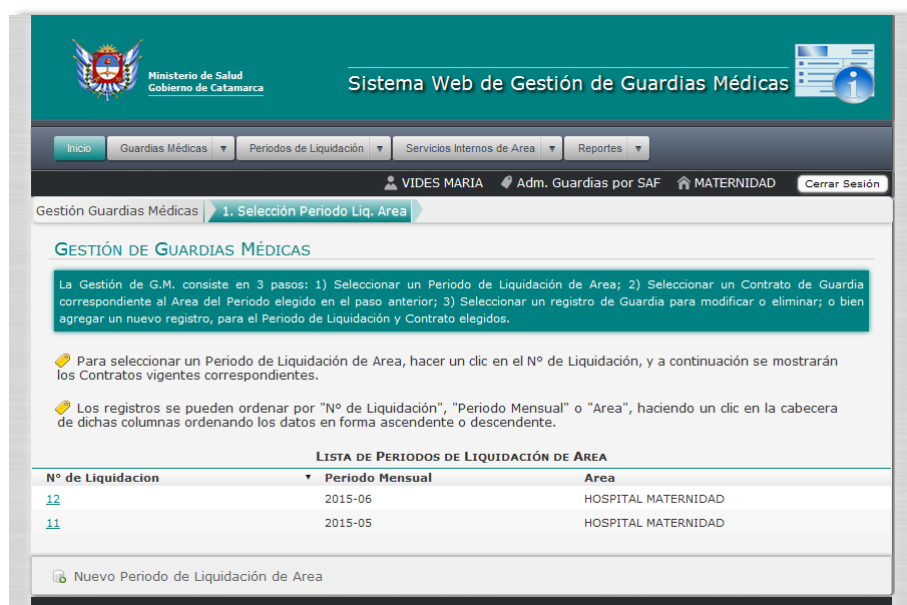


Figura 43: Primera instancia de la Gestión de Guardias Médicas

Haciendo un clic en el link de un “N° de Liquidación” en particular, el sistema presenta la siguiente página, donde se muestra una Lista de Contratos de Guardias según el Área del Periodo seleccionado, tal como se muestra en la Figura 44:

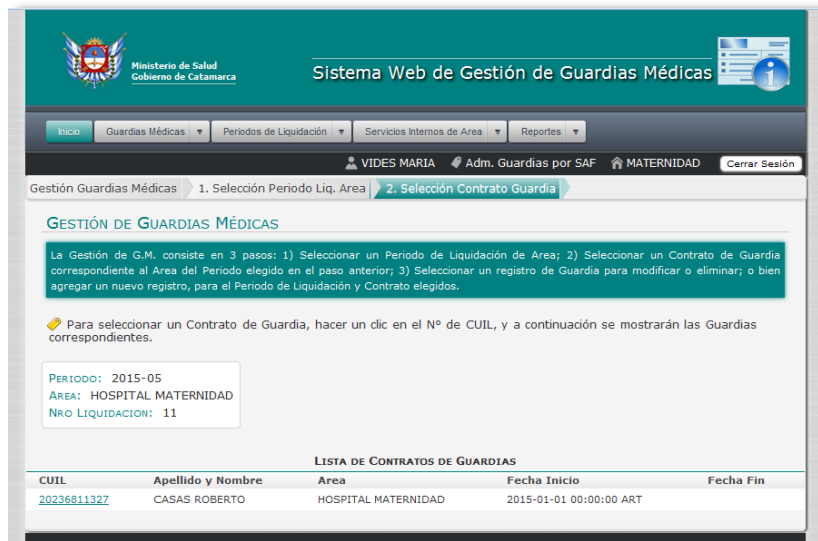


Figura 44: Segunda instancia de la Gestión de Guardias Médicas

De manera similar que la página anterior, se debe seleccionar un Contrato de Guardia de empleado en particular, haciendo un clic en el campo CUIL. A continuación, el sistema presenta finalmente una lista con las guardias médicas registradas pertenecientes al Contrato de Guardia y Período de Liquidación de Área seleccionados como se muestra en la Figura 45:

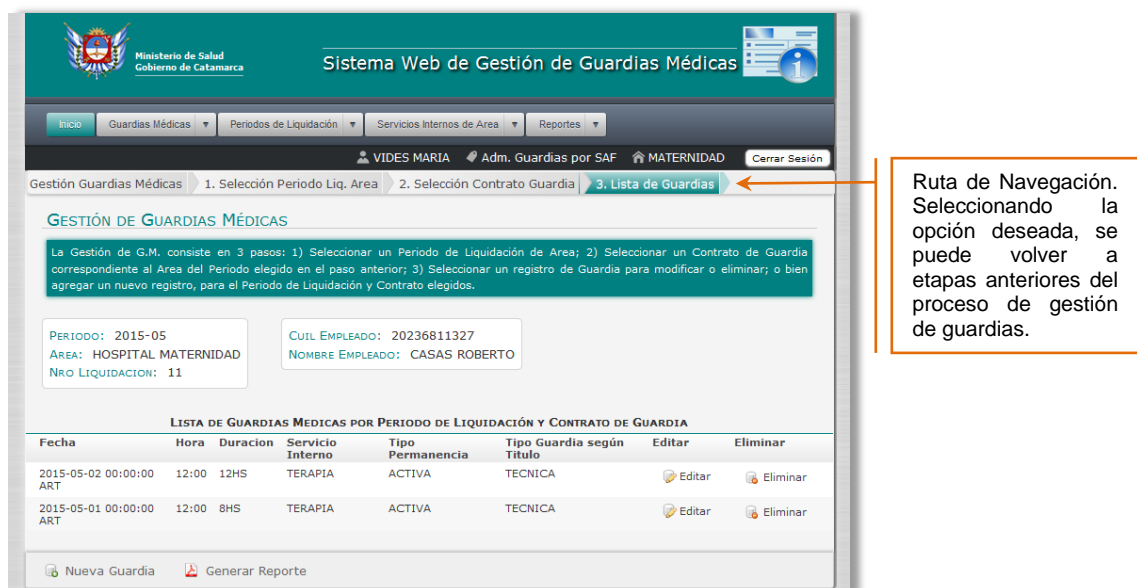


Figura 45: Tercera instancia de la Gestión de Guardias Médicas

Mediante esta página el usuario puede “editar” o “eliminar” un registro de guardia médica en particular, o bien, puede “agregar” una nueva guardia.

El sistema también brinda la opción de generar un reporte en formato “pdf” de las guardias médicas recuperadas para el Contrato de Guardia y Periodo de Liquidación. En la *Sección 5.4.1.4*, se muestra la vista del mencionado reporte.

Si el usuario hace un clic en el botón “Nueva Guardia”, se habilita la siguiente página web para el ingreso de datos (para el caso de edición de guardia se muestra el mismo formulario), tal como se muestra en la Figura 46:

GESTIÓN DE GUARDIAS MÉDICAS - CREACIÓN DE REGISTRO

La Gestión de G.M. consiste en 3 pasos: 1) Seleccionar un Periodo de Liquidación de Área; 2) Seleccionar un Contrato de Guardia correspondiente al Área del Periodo elegido en el paso anterior; 3) Seleccionar un registro de Guardia para modificar o eliminar; o bien agregar un nuevo registro, para el Periodo de Liquidación y Contrato elegidos.

Los campos marcados con asterisco (*) son obligatorios

El Tipo de Guardia (Profesional o Técnica) será determinado en base al Título de máximo nivel alcanzado por el Empleado a la fecha de realización de la Guardia.

PERIODO: 2015-05
ÁREA: HOSPITAL MATERNIDAD
NRO LIQUIDACIÓN: 11

CUIT EMPLEADO: 20236811327
NOMBRE EMPLEADO: CASAS ROBERTO
TÍTULOS EMPLEADO:
* ENFERMERO - Fecha Obtención: 2000-01-01 00:00:00.0 - Tipo Título: TECNICA
* PEDIATRA - Fecha Obtención: 2015-01-01 00:00:00.0 - Tipo Título: PROFESIONAL

FECHA INICIO GUARDIA: 19 enero 2016 *

HORA INICIO (FORMATO 24HS): HH:MM Ejemplo: 23:00 *

DURACION GUARDIA: 8HS *

SERVICIO INTERNO DEL ÁREA: ENFERMERIA *

TIPO PERMANENCIA GUARDIA: ACTIVA *

Guardar Cambios Volver a Inicio

Datos de Periodo de Liquidación de Área y de Contrato de Guardia de empleado

Figura 46: Formulario de modificación o alta de un registro de guardia médica

Como se puede apreciar, todos los campos son obligatorios, por lo que el usuario no debe omitir ningún dato al momento de registrar una guardia. Adicionalmente, la WebApp realiza los siguientes controles al momento de registrar una nueva guardia o una modificación de la misma:

1. La fecha de realización de una guardia debe coincidir con el mes y año del Periodo de Liquidación de Área.

2. Un Empleado no puede realizar más de una guardia del mismo tipo (Activa o Pasiva), para una misma fecha y Área del Contrato.
3. Un Empleado puede realizar guardias en una misma fecha y en varias Áreas, siempre y cuando la suma total de horas no supere las 24 horas, teniendo en cuenta: las horas que el empleado debe cumplir en el día por un cargo en Administración Pública Provincial (si lo tuviere y teniendo en cuenta si el día es Feriado), más las horas de las guardias ya registradas para el empleado, más las horas de la guardia a registrar.

Por ejemplo, si el usuario ingresa una fecha que no coincida con el Periodo de Liquidación, el sistema mostrará la siguiente advertencia (Figura 47):

GESTIÓN DE GUARDIAS MÉDICAS - CREACIÓN DE REGISTRO

La Gestión de G.M. consiste en 3 pasos: 1) Seleccionar un Periodo de Liquidación de Área; 2) Seleccionar un Contrato de Guardia correspondiente al Área del Periodo elegido en el paso anterior; 3) Seleccionar un registro de Guardia para modificar o eliminar; o bien agregar un nuevo registro, para el Periodo de Liquidación y Contrato elegidos.

Los campos marcados con asterisco (*) son obligatorios

El Tipo de Guardia (Profesional o Tecnica) será determinado en base al Título de máximo nivel alcanzado por el Empleado a la fecha de realización de la Guardia.

El mes y año de la fecha ingresada no coincide con el Periodo de Liquidacion 2015-05

PERIODO: 2015-05
 AREA: HOSPITAL MATERNIDAD
 NRO LIQUIDACION: 11

CUIL EMPLEADO: 20236811327
 NOMBRE EMPLEADO: CASAS ROBERTO
 TITULOS EMPLEADO:
 * ENFERMERO - Fecha Obtención: 2000-01-01 00:00:00.0 - Tipo Título: TECNICA
 * PEDIATRA - Fecha Obtención: 2015-01-01 00:00:00.0 - Tipo Título: PROFESIONAL

FECHA INICIO GUARDIA: 19 enero 2016 *

HORA INICIO (FORMATO 24HS HH:MM): 12:00 *

DURACION GUARDIA: 8HS *

Notificación de errores en el ingreso de datos

Figura 47: Notificación de errores en la página web de modificación o alta de registros de guardias médicas

5.4.1.2. Opción de Menú: “Periodos de Liquidación”

Mediante esta opción, se pueden agregar o eliminar registros de Periodos de Liquidación de Área.

En primer lugar la WebApp presenta una página que contiene una lista de registros de Periodos de Liquidación de Área, cuyo estado sea “Abierto” o “Cerrado”. Luego el usuario puede “eliminar” un registro en particular siempre y cuando el estado del mismo no sea “cerrado” y no existan guardias médicas asociadas a él (Ver Figura 48).



Figura 48: Pagina web de Gestión de Periodos de Liquidación de Área

En la Figura 49 se muestra la página web que se presenta en el caso que el usuario haya elegido la opción para agregar un Nuevo registro de Periodo de Liquidación de Área:



Figura 49: Formulario de creación de registro de Periodos de Liquidación de Area

5.4.1.3. Opción de Menú: “Servicios Internos de Área”

Similar a la alternativa anterior, esta opción de menú permite agregar, modificar o eliminar registros de Servicios Internos de un Área en particular.

La WebApp muestra una lista de registros, dando la opción de modificar o eliminar cada uno y también, agregar un nuevo Servicio Interno de Área, tal como se muestra en la Figura 50:



Figura 50: Página web de Gestión de Servicios Internos de Área

Los datos a ingresar para agregar o modificar un registro de esta entidad se muestran en la Figura 51:

The image shows a web application interface for medical shift management. At the top, there is a header with the logo of the 'Ministerio de Salud Gobierno de Catamarca' and the title 'Sistema Web de Gestión de Guardias Médicas'. Below the header is a navigation menu with options like 'Inicio', 'Guardias Médicas', 'Periodos de Liquidación', 'Servicios Internos de Área', and 'Reportes'. A user profile 'VIDES MARIA' and a home button 'MATERNIDAD' are also visible. The main content area is titled 'Gestión de Servicios Internos de Área - CREACIÓN DE REGISTRO' and contains a form with the following elements: a text input field for 'NOMBRE SERVICIO INTERNO DE ÁREA' with a red border and an asterisk, a dropdown menu for 'ÁREA' currently showing 'HOSPITAL MATERNIDAD' with an asterisk, and a checked checkbox for 'ACTIVO'. At the bottom of the form are two buttons: 'Guardar Cambios' and 'Volver a Inicio'. A note above the form states 'Los campos marcados con asterisco (*) son obligatorios'.

Figura 51: Formulario de creación de Servicio Internos de Área

5.4.1.4. Opción de Menú: “Reportes”

Mediante esta opción, se permite al usuario generar 3 tipos de reportes de guardias médicas asociadas a un Periodo de Liquidación de Área en particular.

- **Tipo de Reporte: Detalle de Guardias por Periodo Liq. Área**

El informe que se genera si el usuario elige esta opción es una planilla que contiene el detalle de cada una de las guardias realizadas por cada empleado en un Periodo y Área en particular, identificando si se trata de una guardia Activa o Pasiva, fecha, duración y hora inicio de la misma, y totales de guardias por agente y Área.

Este reporte se incluye en el Expediente que se inicia para el pago de las guardias por Área en el Periodo.

El mencionado informe tiene una vista como la que se muestra en la Figura 52:

Detalle de Guardias Realizadas para el Periodo: 2015-05
 Servicio Administrativo Financiero: MATERNIDAD
 Nº Liquidación D.L.H.: 11
 Denominación Área: HOSPITAL MATERNIDAD

APELLIDO Y NOMBRE	NRO. AGENTE	NRO. CUL	TIPO DE GUARDIA	FECHA DE EFECTIVA REALIZACION DE LA GUARDIA PROFESIONAL/TECNICA																															TOTAL DE GUARDIAS																		
				01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	08	12	16	18	20	24	TOTAL												
CASAS ROBERTO		20226811327	TE ACTIVA		012 12:00																																			0	2	0	0	0	0	2							
CASAS ROBERTO		20226811327	PR PASIVA				08 12:00																																								1	0	0	0	0	0	1
TOTAL DE GUARDIAS POR AREA																																	1	2	0	0	0	0	3														

Figura 52: Vista del Reporte “Detalle de Guardias por Periodo Liq. Área”

▪ **Tipo de Reporte: Resumen de Guardias por Periodo Liq. Área**

La información que se muestra en este tipo de reporte es un resumen de los totales de guardias realizadas, agrupados por Servicios Internos del Área, duración y tipo permanencia de guardias.

El aspecto del reporte generado se muestra en la Figura 53:

Resumen de Guardias Realizadas por Servicio 2015-05
 Servicio Administrativo Financiero: MATERNIDAD
 Nº Liquidación D.L.H.: 11
 Denominación Área: HOSPITAL MATERNIDAD

SERVICIO INTERNO	TOTAL GUARDIAS ACTIVAS							TOTAL GUARDIAS PASIVAS								
	04	08	12	16	18	20	24	TOTAL	04	08	12	16	18	20	24	TOTAL
ENFERMERIA	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1
TERAPIA	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
TOTALES POR AREA	0	0	2	0	0	0	0	2	0	1	0	0	0	0	0	1

Figura 53: Vista del Reporte “Resumen de Guardias por Periodo Liq. Área”

▪ **Tipo de Reporte: Guardias por Agente y Periodo Liq. Area**

El presente reporte se puede generar desde la opción de menú “Reportes” o bien, desde la página de gestión de guardias médicas, en

la instancia donde se muestra la lista de guardias para un Contrato y Periodo de Liquidación de Área en particular.

Desde la opción “Reportes”, se debe elegir la opción “Guardias por Agente y Periodo Liquidación de Área”, y a continuación el sistema habilitará un campo de texto para que el usuario ingrese el N° de CUIL del agente que quiere consultar, tal como se muestra en la Figura 54:

GENERACIÓN DE REPORTES VARIOS DE GUARDIAS MÉDICAS

- Los campos marcados con asterisco (*) son obligatorios
- El Listado de Periodos se puede ordenar por los campos "N° de Liquidación", "Periodo Mensual" y "Area", haciendo un clic en la cabecera de dichas columnas ordenando los datos en forma ascendente o descendente.
- Si elige el Tipo de reporte a generar: "Guardias por Agente y Periodo de Liquidación Area", se habilitará un campo de texto para ingresar el CUIL del Agente a consultar.

SELECCIONE EL TIPO DE REPORTE QUE DESEA GENERAR *

- Detalle de Guardias por Periodo de Liquidación Area
- Resumen de Guardias por Periodo de Liquidación Area
- Guardias por Agente y Periodo de Liquidación Area

INGRESE CUIL DEL AGENTE (SIN GUIONES)*:

Campo de texto para ingresar CUIL del agente para consultar guardias

Figura 54: Selección del tipo de reporte “Guardias por Agente y Periodo Liq. Area”

Cabe aclarar, que el sistema controla que el usuario identificado, según su rol, tenga acceso a la información de guardias del CUIL ingresado.

El reporte generado es el que se muestra en la Figura 55:

guardiasPorAgente-6.pdf - Adobe Reader

Archivo Edición Ver Ventana Ayuda

Herramientas Rellenar y firmar Comentario

1 / 1 139%

Nombre Agente de Guardia: CASAS ROBERTO
N° CUIL: 20236811327
Titulo/s: ENFERMERO - PEDIATRA

Listado de Guardias por Agente para el Periodo: 2015-05
Servicio Administrativo Financiero: MATERNIDAD
Area Programática: HOSPITAL MATERNIDAD
N° Liquidacion D.L.H.: 11

Pagina: 1
Fecha: 20/01/2016

FECHA	HORA	TIPO PERMANENCIA	TIPO GUARDIA SEGUN TITULO	DURACION GUARDIA	SERVICIO INTERNO
02/05/2015	12:00	ACTIVA	TECNICA	12HS	TERAPIA
05/05/2015	12:00	PASIVA	PROFESIONAL	8HS	ENFERMERIA
13/05/2015	15:00	ACTIVA	PROFESIONAL	12HS	ENFERMERIA

Figura 55: Vista del Reporte “Guardias por Agente y Periodo Liq. Area”

5.4.2. Opciones de Menú específicas del Rol Supervisor

El usuario con Rol de Supervisor, tiene acceso a todas las opciones explicadas en la sección anterior, y además se agregan las que se detallan a continuación.

Las opciones de menú para este tipo de usuario se muestran en la Figura 56:

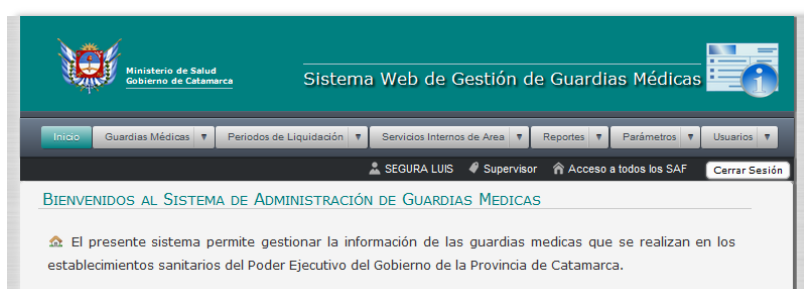


Figura 56: Opciones de menú para el usuario con Rol Supervisor

5.4.2.1. Opción de Menú: “Parámetros ➔ Gestión de Feriados”

Mediante esta opción, el sistema permite gestionar fechas correspondientes a días feriados o no laborables, los cuales se tendrán en cuenta al momento del registro de guardias, controlando la duración máxima de horas que un agente puede realizar en un día.

Al igual que las demás páginas de gestión, se muestra en primer lugar una lista de registros, con la opción de modificar o eliminar cada uno, y también poder agregar un nuevo día feriado.

Cabe aclarar que el sistema no permite eliminar un día feriado, si existen guardias registradas en esa misma fecha.

La Figura 57 presenta la vista de la aplicación web correspondiente a la gestión de feriados.



Figura 57: Página web de Gestión de días feriados

Mediante la página web que se muestra en la Figura 58 se pueden modificar o agregar datos de un día feriado:



Figura 58: Formulario de creación de registro de día feriado

5.4.2.2. Opción de Menú: “Parámetros ➔ Gestión Duración de Guardias”

A través de esta opción se pueden gestionar registros de los bloques de horas correspondientes a una guardia médica. Como en los casos anteriores, se muestra un listado de registros y las opciones para agregar, modificar o eliminar datos (ver Figura 59).

Al igual que los días feriados, el sistema no permite eliminar un registro de duración de guardia, si existen guardias registradas con dicha duración de horas.

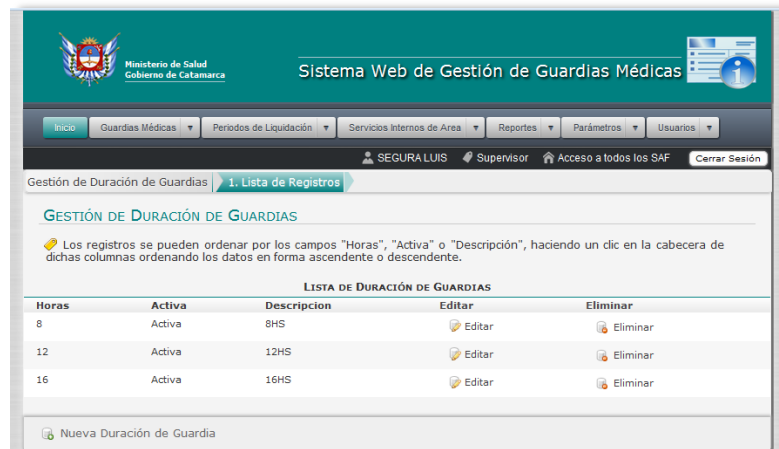


Figura 59: Página web de Gestión de Duración de Guardias

Los datos que solicita el sistema para agregar o modificar una duración de guardia son los que se muestran en el formulario web de la Figura 60:



Figura 60: Formulario de creación de registros de Duración de Guardias

5.4.2.3. Opción de Menú: “Usuarios”

Finalmente, solo el usuario con Rol Supervisor puede gestionar los demás usuarios de la WebApp, siempre con la solicitud y aval de las Autoridades que correspondan.

Los datos solicitados para agregar un nuevo usuario son los que se muestran en el formulario web de la Figura 61:

GESTIÓN DE USUARIOS - CREACIÓN DE REGISTRO

Los campos marcados con asterisco (*) son obligatorios

CUIL *

APELLIDO Y NOMBRE DE USUARIO *

PASSWORD *

EMAIL *

TELEFONO

ROL USUARIO: AGM DE SAF *

S.A.F. DE ACCESO: MATERNIDAD *

CUENTA EXPIRADA

CUENTA BLOQUEADA

HABILITADO

PASSWORD EXPIRADA

[Guardar Cambios](#) [Volver a Inicio](#)

Según el Rol de Usuario seleccionado, se muestra en esta lista desplegable, los SAF o Áreas correspondientes

Figura 61: Formulario de creación de usuario de la WebApp

Al igual que en las demás páginas de gestión de registros, también se puede acceder a una lista de usuario con la opción de modificar o eliminar alguno en particular, tal como se muestra en la Figura 62:

Ministerio de Salud
Gobierno de Catamarca

Sistema Web de Gestión de Guardias Médicas

Inicio Guardias Médicas Períodos de Liquidación Servicios Internos de Área Reportes Parámetros Usuarios

SEGURA LUIS Supervisor Acceso a todos los SAF Cerrar Sesión

Gestión de Usuarios 1. Lista de Registros

GESTIÓN DE USUARIOS DE LA WEBAPP

Los registros se pueden ordenar por "CUIL" o "Nombre Usuario", haciendo un clic en la cabecera de dichas columnas ordenando los datos en forma ascendente o descendente.

CUIL	Nombre Usuario	Email	Telefono	Rol Usuario	Organismo acceso según Rol	Editar	Eliminar
27237846373	VIDES MARIA	mvides@yahoo.com.ar	4459372	AGM DE SAF	MATERNIDAD	Editar	Eliminar
20237966873	SEGURA LUIS	lseg@yahoo.com.ar	4455233	SUPERVISOR	TODOS LOS SAF	Editar	Eliminar
20248099063	LOPEZ JUAN	jlopez@yahoo.com.ar	4433667	AGM DE SAF	ASISTENCIA SANITARIA	Editar	Eliminar
27274837820	VERA ANA	avera@yahoo.com.ar	4495837	AGM DE AREA	AREA PROGRAMATICA 1	Editar	Eliminar

[Nuevo Usuario](#)

Figura 62: Página Web de Gestión de Usuarios de la WebApp

CAPITULO VI - EVALUACION DE LA CALIDAD DEL PROTOTIPO WEB

6.1. INTRODUCCION

Habiendo llegado a esta fase del presente Trabajo Final, se ha culminado con la especificación del modelo de calidad y métricas e indicadores asociados; se ha modelado el prototipo de WebApp y se ha completado el desarrollo del mismo mediante el framework Grails.

Por lo tanto, estamos en condiciones de implementar las restantes actividades de la Estrategia de Medición y Evaluación de Calidad GOCAME, de tal forma que permitan evaluar y determinar en qué grado el prototipo web desarrollado satisface las directrices del Principio “Comprensible” Nivel A de WCAG 2.0, proporcionando información y manejo de interfaces de usuario comprensibles.

La evaluación se realizó en base al modelo de atributos de calidad y al diseño de métricas e indicadores planteados en el Capítulo III del presente trabajo.

Continuando con la clasificación de actividades de GOCAME, en este capítulo se desarrollan las siguientes actividades: (A3) Implementar la medición, (A5) Implementar la Evaluación, y (A6) Analizar y recomendar.

Antes de exponer los resultados de las actividades de implementación de la medición y evaluación, se explicarán algunas consideraciones generales sobre el desarrollo del prototipo web y además, se mostrarán ejemplos del cumplimiento de los atributos de accesibilidad al contenido web.

6.1.1. Consideraciones Generales sobre el desarrollo del Prototipo Web

Como se mencionó anteriormente, para el desarrollo del prototipo de Aplicación Web se utilizó el framework de desarrollo web denominado “Grails”.

Tal como se explicó en la *Sección 2.7.6*, a partir de la definición del modelo de clases de dominio, Grails genera en forma automática las vistas correspondientes (documentos HTML, CSS) de acuerdo a la funcionalidad de “crear, leer, actualizar y eliminar (CRUD) entidades”. La generación de estas vistas se basa en una plantilla que el desarrollador puede modificar y preestablecer, de tal forma que todas las páginas web generadas respondan a este formato. Es decir, que todos los documentos HTML generados cuentan con un diseño y formato uniforme, basados en un esquema común, y respetando el diseño plasmado en el Modelo de Presentación de la *Sección 4.3.4*.

Las páginas web generadas por el framework corresponden a la versión 5 de HTML, la cual constituye la última Recomendación de W3C a partir del 28 de Octubre de 2014²⁶.

Entre las nuevas características que brinda HTML 5, se encuentra la posibilidad de realizar validaciones básicas de campos de un formulario, mediante el navegador web, sin la necesidad de convocar funciones del lenguaje de scripting “JavaScript”.

Cuando se define un modelo de clases de dominio en Grails, se pueden definir restricciones sobre los atributos de dichas clases como por ejemplo, que cierto atributo no sea nulo (obligatorio), que tenga un número máximo de caracteres, o que corresponda a algún tipo de dato en particular. Estas restricciones se toman en cuenta en el momento en que se generan los documentos HTML, adecuando la definición de las

²⁶ Especificación de HTML5 en: <https://www.w3.org/TR/html/>

etiquetas <html> correspondientes a cada atributo de las clases de dominio.

En definitiva, los campos o atributos, se validan en forma básica tanto desde el navegador mediante HTML 5 y luego en el servidor, de acuerdo a restricciones más específicas.

6.2. EJEMPLOS DEL CUMPLIMIENTO DE ATRIBUTOS DE ACCESIBILIDAD WEB

Con el fin de ejemplificar y visualizar el cumplimiento de las características de accesibilidad al contenido web planteadas para este trabajo, se expondrá de qué manera en concreto, por cada atributo del Árbol de Requerimientos de Calidad, el Prototipo Web desarrollado satisface tales requerimientos. Para una mejor comprensión, se debe tener en mente la definición de cada atributo y las métricas involucradas.

▪ Atributo 1.1.1 - Correcta identificación del idioma de las páginas web mediante software

El idioma general de la aplicación web es el “Español”, por lo tanto, en la plantilla principal, de la cual se basan las vistas generadas por el framework, se identifica el idioma mediante el atributo “lang” y su codificación “es” correspondiente al lenguaje español:

```
<html lang="es" class="no-js">
```

▪ Atributo 1.2.1 - Inexistencia de cambios de contexto cuando componentes reciben el foco

Para verificar el cumplimiento de este atributo se requiere de la observación de la aplicación en funcionamiento, por lo tanto no se puede demostrar esta característica en este apartado. De todos modos,

se tuvo en cuenta esta restricción, para no iniciar cambios de contexto cuando componentes reciben el foco.

- **Atributo 1.2.2 - Inexistencia de cambios de contexto al modificar estado de componentes**

Al igual que la situación anterior, se requiere de la observación de la aplicación en funcionamiento para verificar este atributo. Sin embargo, se puede adelantar que no se inician cambios de contexto al modificar estado de componentes, sin seguir las recomendaciones adecuadas.

- **Atributo 1.2.3 - Cambios de contexto generados apropiadamente**

Durante el desarrollo de la webapp, se presentaron situaciones donde se requirieron inicios de cambios de contextos, no obstante, se tuvieron en cuenta las formas adecuadas de realizar los mismos.

Por ejemplo, en la página web de “Generación de Reportes Varios de Guardias Médicas” que se muestra en la Figura 63, se solicita al usuario que seleccione el *Tipo de Reporte* que desea generar, brindando tres opciones a través de un botón de radio:

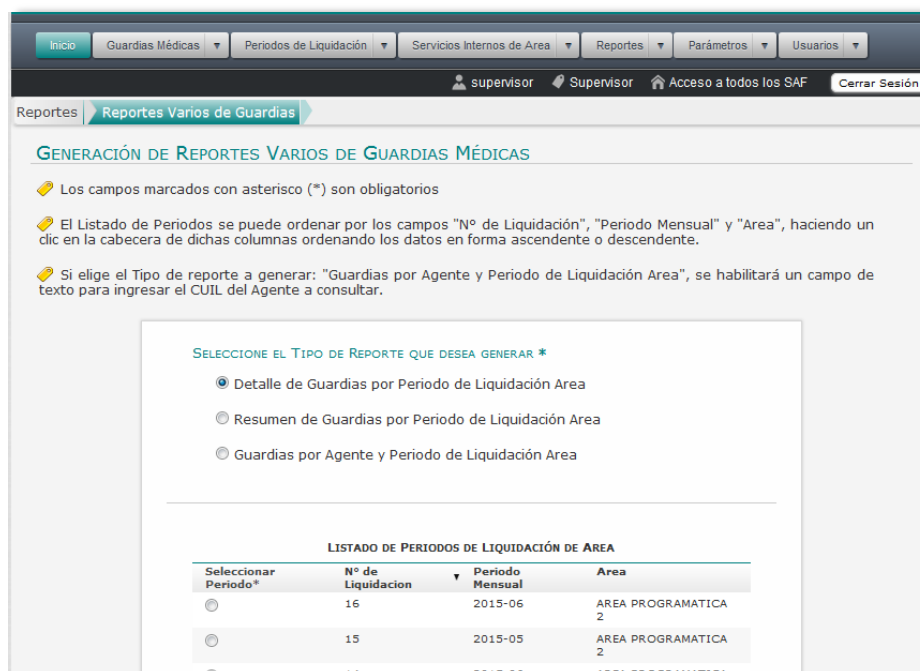


Figura 63: Página web de Generación de Reportes Varios

Si el usuario elige la opción “Guardias por Agente y Periodo de Liquidación Área”, se habilita una etiqueta y campo de texto para ingresar el CUIL del empleado. Esta funcionalidad se considera un cambio de contexto, que se inicia al modificar el estado de un botón de radio, no obstante, esta acción fue realizada directamente por el usuario y además, en la sección informativa de la página web, se ha informado previamente sobre esta situación, por lo tanto se considera una manera adecuada de iniciar el mencionado cambio de contexto. La Figura 64 muestra el estado de la página web luego de haber iniciado el cambio de contexto:

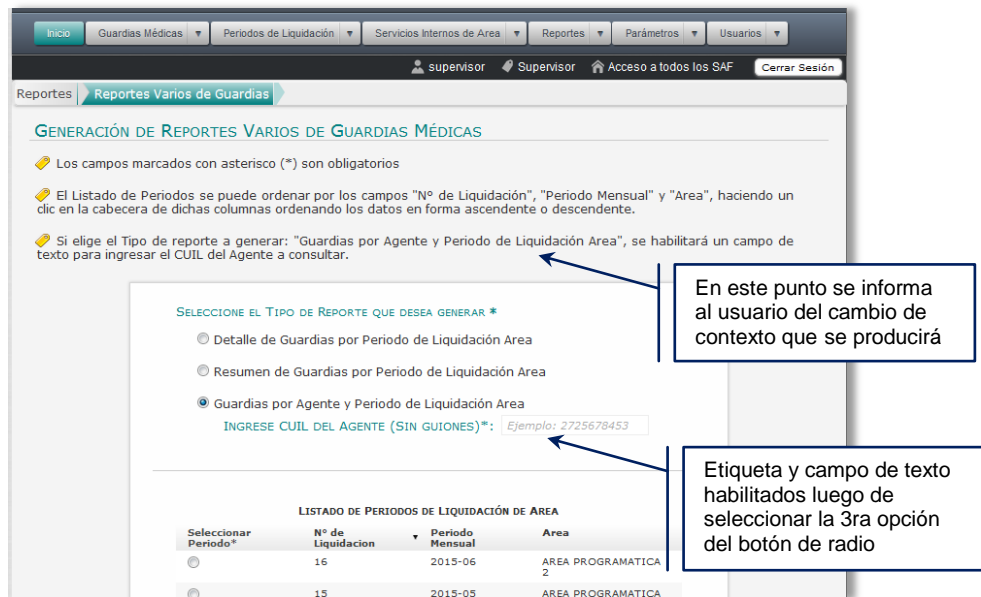


Figura 64: Página web de Generación de Reportes Varios luego de haber iniciado cambio de contexto

▪ **Atributo 1.3.1.1 - Uso de notificación textual de error para campos obligatorios**

Cuando el sistema solicita el ingreso de datos que son obligatorios y el usuario no los completa, en primer lugar se ejecuta la validación de HTML 5, notificando textualmente y señalando el campo que se debe completar. En la Figura 65 se muestra esta situación:

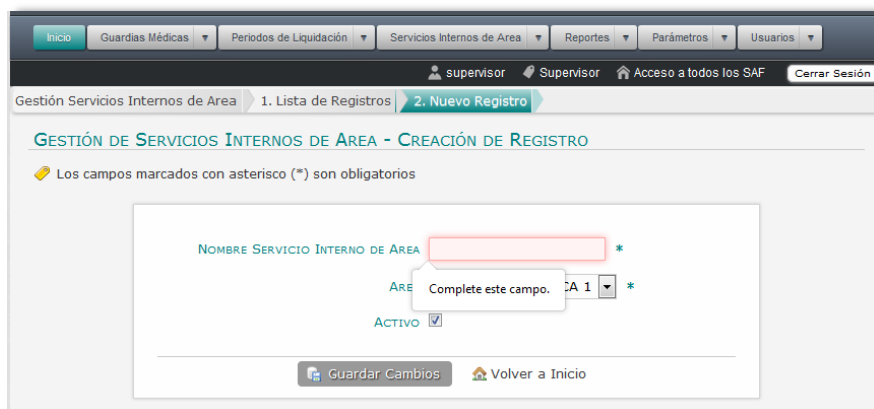


Figura 65: Notificación textual de error de campo obligatorio desde el navegador web

En el caso de que por algún motivo no se haya realizado la validación mediante HTML 5 y el navegador web; y los datos del formulario fueran enviados a través del botón “submit” al Servidor Web, la validación de datos se realiza desde la WebApp conforme a las restricciones de los atributos definidas en el modelo de dominio. En la Figura 66 se puede apreciar una notificación textual de error por haber omitido un campo obligatorio y a la vez, como en el caso anterior, se identifica a dicho campo resaltando su borde y fondo en color rojo:

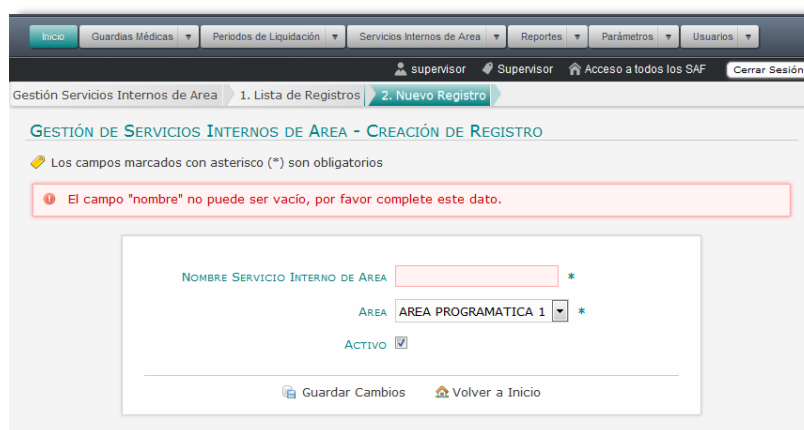


Figura 66: Notificación textual de error de campo obligatorio desde la WebApp

▪ **Atributo 1.3.1.2 - Notificación textual de error apropiada para campos con determinado formato**

El formato requerido para un cierto atributo se puede definir como restricción en la clase de dominio del sistema, lo que luego permite al framework Grails trasladar la verificación al momento de recibir los datos asociados desde un formulario web.

Por ejemplo, en la definición de la Clase “PeriodoLiquidacionArea”, se especifica que el atributo “periodoMensual” debe tener un formato de 4 dígitos numéricos para el año y 2 dígitos para el mes (aaaamm), sin mediar ningún signo entre ellos. Además, se determina que el valor correspondiente a un mes debe estar entre 01 y 12. En la Figura 67 se

muestra una fracción de la sección de definición de restricciones para esta clase:

```
class PeriodoLiquidacionArea {
    ...
    static constraints = {
        periodoMensual (maxSize:6, matches: "[0-9]{4}[0]{1}[1-9]{1}|[0-9]{4}[1]{1}[0-2]{1}",
            unique: ['area'])
        ...
    }
}
```

Figura 67: Ejemplo de definición de formato de atributo en una clase de dominio

Con solo esta definición del atributo, el framework Grails verifica que el dato asociado a este campo satisfaga la restricción, caso contrario, se muestra la siguiente notificación textual de error cuando no se ingresó correctamente el formato requerido, además de brindar un ejemplo para ayudar a solucionar el problema, tal como se muestra en la Figura 68:



Figura 68: Notificación textual de error para campo con determinado formato

En general, para toda la webapp se implementó esta manera de trabajar, personalizando los mensajes de error que emite el framework, procurando que sean lo más específicos posibles y brindando ejemplos de datos correctos en cada situación.

- **Atributo 1.3.2.1 - Representatividad de texto de etiquetas**

Las etiquetas de texto de los controles de formularios, se generan automáticamente desde el framework en el momento que se crean las vistas correspondientes, de acuerdo a los nombres de los atributos definidos en las clases de dominio. Sin embargo, se ha verificado y en el caso que fue necesario, se ha modificado el texto de las etiquetas para que sean lo más representativas posible del campo al que se refieren.

- **Atributo 1.3.2.2 - Identificación adecuada de controles de formularios con determinado formato**

Para el caso de campos de datos que requieren de un formato específico, se ha diseñado la webapp de tal manera que se brinda información sobre el mismo, dentro de la etiqueta <label> correspondiente a cada campo de entrada de datos. Además, mediante el atributo “placeholder” de la etiqueta <input> de HTML 5, se proporciona un ejemplo del valor esperado. A continuación, en la Figura 69 se muestran algunos ejemplos de etiquetas y campos de textos que requieren un formato en particular:

Figura 69: Identificación de campos con un determinado formato

- **Atributo 1.3.2.3 - Identificación adecuada de campos obligatorios**

Dentro de la WebApp, los campos obligatorios se identifican mediante un asterisco que se ubica en el lado derecho del campo, y a su vez, en

la sección informativa de cada página, se explica al usuario el significado de dicho asterisco. En la Figura 70 se puede apreciar un ejemplo de este atributo:

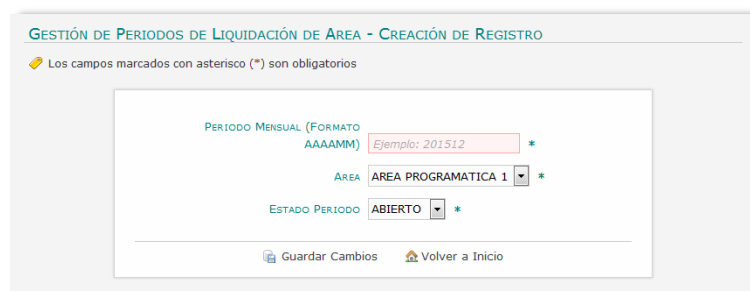


Figura 70: Ejemplo de identificación de campos obligatorios

- **Atributo 1.3.2.4 - Asociación adecuada de etiquetas con controles de formularios**

Todos los campos de formularios de la WebApp tienen su respectiva etiqueta `<label>` asociada mediante el atributo “for” al control que corresponda. En la Figura 71 se puede observar la vista de una página web y la pantalla de inspección de código fuente de la misma, donde se muestra la etiqueta `<label>` para “Cantidad de horas...” y su vinculación con el campo “horas” a través del atributo “for” de dicha etiqueta:

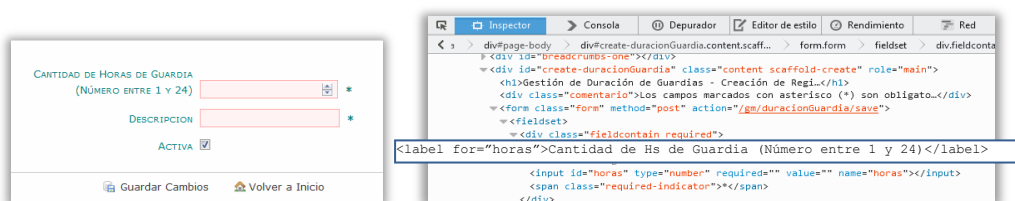


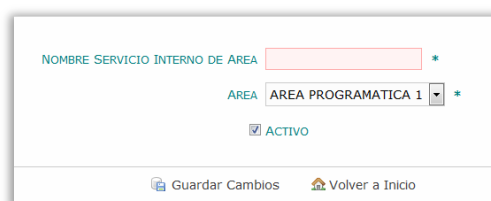
Figura 71: Ejemplo de asociación de etiquetas con controles de formularios

- **Atributo 1.3.2.5 - Etiquetas de texto de controles de formularios ubicadas adecuadamente**

La ubicación adecuada de las etiquetas respecto a los controles de formularios es inmediatamente *antes* del campo si se trata de un

campo de texto, lista desplegable, etc., en otro caso, si se tiene un botón de radio o casilla verificación, la etiqueta debe encontrarse inmediatamente *después* del control.

En la Figura 72 se muestra la ubicación adecuada de las etiquetas según se trate de un campo de texto, lista desplegable o casilla de verificación:



The screenshot shows a web form with three input elements. The first is a text input field with the label 'NOMBRE SERVICIO INTERNO DE AREA' positioned to its left. The second is a dropdown menu with the label 'AREA' positioned to its left and the selected value 'AREA PROGRAMATICA 1' inside the dropdown. The third is a checked checkbox with the label 'ACTIVO' positioned to its left. At the bottom of the form, there are two buttons: 'Guardar Cambios' and 'Volver a Inicio'.

Figura 72: Ejemplo de ubicación de etiquetas de campo de texto, lista desplegable y casilla de verificación

6.3. (A3) IMPLEMENTACIÓN DE LA MEDICIÓN

El proceso de medición consiste en obtener un valor o medida, numérica o categórica, para cada uno de los atributos de una o más entidades, utilizando como entrada el conjunto de métricas obtenidas en el diseño de la medición (Sección 3.3)

Finalmente, se implementó en esta etapa la medición de los atributos del árbol de requerimientos, aplicando los métodos de cálculo o procedimientos de medición definidos en cada métrica correspondiente.

Se destaca que esta fase del trabajo se realizó en forma iterativa junto con el desarrollo y diseño de la webapp, ya que al momento de la medición, se verificó que en algunos casos no se cumplía con los requisitos de accesibilidad, por lo que fue necesario, ajustar y corregir la webapp y volver a realizar la medición de atributos.

A continuación, en la Tabla 10 se muestran los valores de los atributos obtenidos en la medición del prototipo web desarrollado:

Tabla 10: Resultados de la medición de atributos de calidad

Atributo del Modelo de requerimientos	Métrica indirecta asociada	Valor Métrica
1.1.1. Correcta identificación del idioma de las páginas web mediante software	Porcentaje de páginas web con identificación de idioma mediante software (%PIHTML)	$\%PIHTML = \frac{22}{22} * 100 = 100\%$
1.2.1. Inexistencia de cambios de contexto cuando componentes reciben el foco	Porcentaje de inexistencia de cambios de contexto cuando elementos reciben el foco (%ICCF)	$\%ICCF = \left(1 - \frac{0}{9}\right) * 100 = 100\%$ $\#TCC = 0 + 0 + 9$
1.2.2. Inexistencia de cambios de contexto al modificar estado de componentes	Porcentaje de inexistencia de cambios de contexto al modificar estado de componentes (%ICCM)	$\%ICCM = \left(1 - \frac{0}{9}\right) * 100 = 100\%$ $\#TCC = 0 + 0 + 9$
1.2.3. Cambios de contexto generados apropiadamente	Porcentaje de cambios de contextos apropiados (%CCA)	$\%ICCF = \left(\frac{9}{9}\right) * 100 = 100\%$ $\#TCC = 0 + 0 + 9$
1.3.1.1. Uso de notificación textual de error para campos obligatorios	Porcentaje de notificación textual de error para campos obligatorios (%NTCO)	$\%NTCO = \frac{43}{43} * 100 = 100\%$
1.3.1.2. Notificación textual de error apropiada para campos con determinado formato	Porcentaje de notificación textual de error para campos con formato (%NTCF)	$\%NTCF = \frac{8}{8} * 100 = 100\%$
1.3.2.1. Representatividad de texto de etiquetas	Porcentaje de etiquetas representativas (%ER)	$\%ER = \frac{31}{31} * 100 = 100\%$
1.3.2.2. Identificación adecuada de controles de formularios con determinado formato	Porcentaje de identificación adecuada para campos con formato (%IACF)	$\%IACF = \frac{8}{8} * 100 = 100\%$
1.3.2.3. Identificación adecuada de campos obligatorios	Porcentaje de identificación adecuada para campos obligatorios (%IACO)	$\%IACO = \frac{43}{43} * 100 = 100\%$
1.3.2.4. Asociación adecuada de etiquetas con controles de formularios	Porcentaje de asociación adecuada entre etiqueta y control de formulario (%AAECF)	$\%AAECF = \frac{31}{31} * 100 = 100\%$

Atributo del Modelo de requerimientos	Métrica indirecta asociada	Valor Métrica
1.3.2.5. Etiquetas de texto de controles de formularios ubicadas adecuadamente	Porcentaje de etiquetas ubicadas adecuadamente (%EUA)	$\%EUA = \frac{31}{31} * 100 = 100\%$

6.4. (A5) IMPLEMENTACIÓN DE LA EVALUACION

El objetivo de esta actividad es obtener los valores de los distintos indicadores, tanto elementales (IE) como parciales y globales (IP e IG respectivamente), utilizando la especificación obtenida en el proceso de diseño de la evaluación; el árbol de requerimientos y los valores obtenidos al implementar o ejecutar la medición. De esta forma podremos saber en qué proporción se satisfacen los requerimientos establecidos en la necesidad de información.

Cabe aclarar que para obtener una conformidad con las WCAG 2.0, se deben considerar en el análisis todas las páginas y sus elementos pertenecientes al sitio o aplicación web, es decir, que la conformidad no se puede lograr si una página o parte de la misma se excluye.

En la Tabla 11 se expone el resultado final de la evaluación, donde se puede apreciar que el Prototipo web desarrollado satisface en su totalidad el concepto de calidad “Accesibilidad en la comprensión de la Información y del manejo de IU”, cumpliendo de esta manera con el objetivo propuesto en el marco del presente Trabajo Final.

Tabla 11: Resultados de la evaluación de calidad del prototipo de aplicación web

Concepto/Subconcepto/Atributo	IE	IP/IG
1. Accesibilidad en la comprensión de la Información y del manejo de IU		100%
1.1. Legibilidad de contenidos textuales		100%
1.1.1. <i>Correcta identificación del idioma de las páginas web mediante software</i>	100%	

Concepto/Subconcepto/Atributo	IE	IP/IG
1.2. Predictibilidad del contenido		100%
1.2.1. <i>Inexistencia de cambios de contexto cuando componentes reciben el foco</i>	100%	
1.2.2. <i>Inexistencia de cambios de contexto al cambiar estado de componentes</i>	100%	
1.2.3 <i>Cambios de contexto generados apropiadamente</i>	100%	
1.3. Asistencia en el ingreso de datos en formularios		100%
1.3.1. Identificación de errores apropiada		100%
1.3.1.1. <i>Uso de notificación textual de error para campos obligatorios</i>	100%	
1.3.1.2. <i>Notificación textual de error apropiada para campos con determinado formato</i>	100%	
1.3.2. Identificación de componentes de formularios apropiada		100%
1.3.2.1. <i>Representatividad de texto de etiquetas</i>	100%	
1.3.2.2. <i>Identificación adecuada de controles de formulario con determinado formato</i>	100%	
1.3.2.3. <i>Identificación adecuada de campos obligatorios</i>	100%	
1.3.2.4. <i>Asociación adecuada de etiquetas con controles de formularios</i>	100%	
1.3.2.5. <i>Etiquetas de texto de controles de formularios ubicadas adecuadamente</i>	100%	

6.5. (A6) ANALISIS Y RECOMENDACION

De acuerdo a los resultados de la evaluación, se puede observar que el grado de cumplimiento global del concepto “Accesibilidad en la comprensión de la Información y del manejo de IU” es del 100% para el prototipo SiWebGM, ya que la evaluación de los indicadores elementales y parciales de todos los atributos y subconceptos del árbol de requerimientos arrojaron resultados completamente satisfactorios. Esto fue posible porque desde un inicio del ciclo de desarrollo de la webapp se planteó la necesidad de cumplir con los atributos de accesibilidad al

contenido web, definidos en el modelo de calidad. Más aun, al momento de realizar el modelado y posterior desarrollo de la webapp, ya se conocía la forma en que se medirían y evaluarían dichos atributos, por lo cual se volcó ese conocimiento e información en las actividades mencionadas.

Por lo tanto, se concluye que la evaluación del prototipo web ha resultado en el máximo nivel de satisfacción posible, cumpliendo plenamente con las características de accesibilidad al contenido web relacionadas con la comprensión de la información y del manejo de interfaces de usuario, según el nivel A de las Pautas de Accesibilidad al Contenido Web versión 2.0.

CAPITULO VII – CONCLUSIONES Y TRABAJOS FUTUROS

7.1. CONCLUSIONES FINALES

El abordaje del problema de gestión de Guardias Médicas mediante una propuesta de solución basada en una aplicación web resultó viable tanto desde el punto de vista conceptual como funcional.

El sistema web desarrollado ha permitido automatizar el proceso de gestión de guardias médicas, reduciendo las actividades que anteriormente se realizaban en forma manual, brindando un servicio más ágil, integral, confiable, y libre de inconsistencias, ya que se realizan todo tipo de controles internos al momento de registrar una guardia, para todas las Áreas de la provincia en forma simultánea.

La utilización de la metodología UWE para realizar el modelado del sistema web propuesto permitió identificar los requerimientos funcionales y las reglas de negocio pertinentes, como así también, facilitó la definición del Modelo de Casos de Uso detallando las diferentes secuencias de interacción de los usuarios con la aplicación web. Los modelos Conceptual, Navegación y Presentación del prototipo SiWebGM esquematizan la configuración de la interfaz de organización y presentación de la información del sistema web propuesto.

La experiencia de aplicación de UWE en una situación problemática particular posibilita valorar positivamente la metodología ya que brinda una serie de modelos de un sistema web que son relevantes para tener una visión global del contexto de interoperabilidad y la correspondiente definición y especificación de la forma de procesamiento de datos, navegación y presentación de la información. También fue posible incluir los aspectos de accesibilidad web que debía satisfacer la webapp, desde

la etapa de modelado, más específicamente, en el modelo de presentación, con lo cual, los modelos de UWE resultantes no solo facilitaron y optimizaron el desarrollo del sistema, sino que también se obtuvo un producto con atributos de calidad incorporados desde un comienzo a la WebApp.

Por otra parte, el prototipo SiWebGM fue desarrollado con herramientas Opensource (gratuitas y de código abierto) como ser: Netbeans, Postgresql, Grails, etc., permitiendo así una reducción significativa en el costo total de producción de la aplicación. A su vez, el uso del framework Grails y la incorporación de plugins (complementos), como el de seguridad, permitió desarrollar funcionalidades que redujeron significativamente el tiempo de desarrollo. Su facilidad de uso aminora la curva de aprendizaje tanto a los desarrolladores novatos como expertos, apreciando la necesidad de desarrollar de forma ágil mediante la automatización de tareas repetitivas y de bajo nivel.

Respecto al Área de evaluación de calidad del software, en este trabajo hemos analizado las Pautas de Accesibilidad al Contenido Web 2.0, las cuales consideramos que constituyen una muy buena orientación para plantear un modelo de calidad, mediante la construcción de un árbol de requerimientos no funcionales, determinando las características y sub-características necesarias para poder calcular el nivel o grado de cumplimiento de la Accesibilidad al contenido web.

Por cada pauta y criterio de conformidad de las WCAG 2.0 se asocian técnicas suficientes para satisfacer dichos criterios de conformidad según el nivel de accesibilidad requerido. A su vez, generalmente, cada técnica suficiente se relaciona con un ítem y/o evento de bajo nivel de abstracción de la aplicación web, lo cual nos ha permitido traducirlas como uno o varios atributos cuantificables (que puede ser medido directamente) de nuestro árbol de requerimientos. Cabe aclarar, que para el desarrollo del

presente trabajo, se definió un conjunto de atributos que permitan evaluar el cumplimiento del Principio 3 “Compresible”, Nivel A de WCAG 2.0, tomando en cuenta las técnicas suficientes para la tecnología HTML.

La cuantificación de los atributos se realizó mediante el diseño e implementación de métricas e indicadores, conforme a la estrategia GOCAME, la cual posee un conjunto robusto de especificaciones de métodos, procesos y marco conceptual que consideramos apropiados para aplicar en este contexto de análisis.

7.2. TRABAJOS FUTUROS

En lo que se refiere al dominio del problema de la Aplicación Web desarrollada, en un futuro se pueden implementar nuevas funcionalidades a la misma, como ser: gestión de licencias de empleados con contratos de guardia, gestión de auditorías de usuario y generación de otros reportes de control para las áreas de auditoría de gobierno.

Así también, sería conveniente que en sucesivas versiones de la WebApp, se incorpore la funcionalidad completa del “*Sistema de Gestión de Contratos por Guardias Médicas*”, de tal forma que exista una sola aplicación web unificada respecto de la gestión total de los contratos de guardias y las guardias efectivamente realizadas. Esto beneficiaría en gran medida las tareas de mantenimiento de los sistemas software del gobierno, y la integración y unificación de datos y tecnologías.

Por otra parte, respecto a la evaluación de la accesibilidad web, se puede avanzar en la especificación del modelo de calidad, incluyendo las restantes pautas y criterios de conformidad que integran la norma WCAG 2.0, de tal manera de contar con un modelo integral de evaluación de la accesibilidad al contenido web.

De igual manera, se puede definir un nuevo modelo de agregación de preferencias para los indicadores parciales y globales, incorporando otras

estructuras de agregación de preferencias compuestas por diferentes funciones generalizadas de conjunción/disyunción del método LSP (Logical Scoring of Preference).

En el presente trabajo se utilizó la función de conjunción/disyunción denominada “Media Aritmética” y se asignaron pesos equivalentes entre atributos correspondientes a un mismo concepto o subconcepto, ya que se consideró que los mismos tienen igual importancia. Sin embargo, se podría realizar una clasificación de los atributos en obligatorios, deseables u opcionales, de acuerdo a las técnicas suficientes o complementarias de WCAG 2.0, asignando diferentes pesos a dichos atributos y utilizando otras funciones de conjunción/disyunción que permitan obtener un cálculo más preciso en base a los criterios elementales.

Otra línea de trabajo que se puede continuar en un futuro es el desarrollo de una herramienta software para gestionar los datos de métricas e indicadores recopilados durante la medición y evaluación de sistemas software. Así también, se podría avanzar sobre el desarrollo de una herramienta que permita medir y evaluar los aspectos objetivos de la accesibilidad web y que no necesitan de una revisión humana, en base a una definición de un modelo de atributos de calidad y métricas automatizadas.

ANEXOS

ANEXO A – ESPECIFICACION DE METRICAS DE CALIDAD

Concepto 1: Accesibilidad en la comprensión de la Información y del manejo de IU

Sub-Concepto 1.1: Legibilidad de contenidos textuales

Atributo 1.1.1. Correcta identificación del idioma de las páginas web mediante software			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p><i>Nombre:</i> Porcentaje de páginas web²⁷ con identificación de idioma mediante software (%PIHTML)</p> <p><i>Tipo:</i> Indirecta</p> <p><i>Métricas relacionadas:</i> 1) Número de páginas web que poseen el atributo lang dentro del elemento html (#PALHTML) 2) Total de páginas web (#TPW)</p>	<p>Determinar el porcentaje entre el número de páginas web que posean una identificación del idioma mediante el atributo lang dentro del elemento html y el número total de páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%PIHTML = \frac{\#PALHTML}{\#TPW} * 100$	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acronimo: %
<p><i>Nombre:</i> Número de páginas web que poseen el atributo lang dentro del elemento html (#PALHTML)</p> <p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Páginas web con identificación de idioma mediante atributo lang del elemento html</p>	<p>Determinar el número de páginas web que posean una identificación del idioma mediante atributo lang dentro del elemento html.</p>	<p>Especificación de método de medición: Inicializar la variable <i>#PALHTML = 0</i>. Por cada documento html diferente generado por la aplicación web, dentro del Rol de Supervisor, verificar que:</p> <ol style="list-style-type: none"> 1) el elemento "html" contiene el atributo "lang" y/o el atributo "xml:lang" 2) el valor de lang es válido y corresponde a algún idioma en particular según la codificación establecida en ISO 639-2 y BCP 47. 3) el lenguaje del texto en general de la página web coincide con el código del lenguaje especificado. <p>Si se cumplen todas las condiciones</p>	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: páginas web con identificación de idioma

²⁷ Una página web es la representación visual de un documento HTML, por lo que mediante un navegador web se puede ver el código fuente de la misma.

		anteriores, entonces #PALHTML = #PALHTML + 1 Si no se cumplen alguna de las condiciones anteriores, no sumar ningún valor a #PALHTML. Tipo de método de medición: objetivo	
<i>Nombre:</i> Total de páginas web (#TPW) <i>Tipo:</i> Directa Atributo relacionado: Páginas web de la WebApp	Determinar el número total de páginas (documentos html) de la aplicación web.	Especificación de método de medición: Inicializar la variable #TPW = 0. Ingresar a la URL principal de la webapp, e identificarse con el Rol de usuario "Supervisor". Luego navegar por todas opciones de menú y otras opciones que provea la aplicación, agregando 1 (uno) a #TPW por cada página web diferente (documento html) generada. Nota: No se sumarán las páginas correspondientes a paginación de resultados. Tipo de método de medición: objetivo	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: Página web

Sub-Concepto 1.2: Predictibilidad del contenido

Atributo 1.2.1. Inexistencia de cambios de contexto cuando componentes reciben el foco			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<i>Nombre:</i> Porcentaje de inexistencia de cambios de contexto cuando elementos reciben el foco (%ICCF) <i>Tipo:</i> Indirecta <i>Métricas relacionadas:</i> 1) Número de cambios de contexto generados cuando componentes reciben el foco (#CCF) 2) Total de cambios de contextos generados (#TCC)	Determinar el porcentaje del complemento entre el número de cambios de contexto generados cuando componentes reciben el foco y el número total de cambios de contexto del conjunto de páginas web de la aplicación.	Especificación de Fórmula: $\%ICCF = \left(1 - \frac{\#CCF}{\#TCC}\right) * 100$ Si #TCC = 0 entonces %ICCF = 100 ya que al no presentarse ningún cambio de contexto en el conjunto de páginas web, tampoco existen cambios de contexto cuando componentes reciben el foco	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %
<i>Nombre:</i> Número de cambios de contexto generados cuando componentes reciben el foco (#CCF)	Determinar el número de cambios de contexto generados cuando componentes reciben el foco, dentro del conjunto de páginas web de la	Especificación de método de medición: Inicializar la variable #CCF = 0. Por cada página web generada por la aplicación, utilizar un teclado y presionar la tecla "tab" para pasar el foco a cada elemento o componente	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta

<p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Cambios de contexto generados cuando componentes reciben el foco</p>	aplicación.	<p>de la página.</p> <p>Si se genera algún cambio de contexto cuando un elemento reciba el foco, y el mismo no fue informado con anterioridad al usuario, entonces #CCF = #CCF + 1</p> <p>Tipo de método de medición: objetivo</p>	<p>▪ Unidad: cambios de contexto generados cuando componentes reciben el foco</p>
<p><i>Nombre:</i> Total de cambios de contextos generados (#TCC)</p> <p><i>Tipo:</i> Indirecta</p> <p><i>Métricas relacionadas:</i> 1) Número de cambios de contexto generados cuando componentes reciben el foco (#CCF) 2) Número de cambios de contexto generados cuando se modifica configuración de componentes (#CCM) 3) Número de cambios de contexto generados apropiadamente (#CCA)</p>	Determinar el número total de cambios de contextos generados en el conjunto de páginas web de la aplicación.	<p>Especificación de Fórmula: $\#TCC = \#CCF + \#CCM + \#CCA$</p> <p>Tipo de método de medición: objetivo</p>	<p>▪ Representación: Discreta</p> <p>▪ Tipo de valor: Entero</p> <p>▪ Tipo de escala: Absoluta</p> <p>▪ Unidad: cambios de contextos de una página web</p>

Atributo 1.2.2. Inexistencia de cambios de contexto al modificar estado de componentes

Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p><i>Nombre:</i> Porcentaje de inexistencia de cambios de contexto al modificar estado de componentes (%ICCM)</p> <p><i>Tipo:</i> Indirecta</p> <p><i>Métricas relacionadas:</i> 1) Número de cambios de contexto generados cuando se modifica el estado de componentes (#CCM) 2) Total de cambios de contextos generados (#TCC)</p>	Determinar el porcentaje del complemento entre el número de cambios de contexto generados cuando se modifica el estado de componentes y el número total de cambios de contexto del conjunto de páginas web de la aplicación.	<p>Especificación de Fórmula: $\%ICCM = \left(1 - \frac{\#CCM}{\#TCC}\right) * 100$</p> <p>Si #TCC = 0 entonces %ICCM = 100 ya que al no presentarse ningún cambio de contexto en el conjunto de páginas web, tampoco existen cambios de contexto cuando se modifica el estado de los componentes de las páginas web.</p>	<p>▪ Representación: Continua</p> <p>▪ Tipo de valor: Real</p> <p>▪ Tipo de escala: Proporción</p> <p>▪ Unidad: Porcentaje</p> <p>▪ Acronimo: %</p>
<p><i>Nombre:</i> cambios de contexto generados cuando se modifica el estado de componentes (#CCM)</p> <p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i></p>	Determinar el número de cambios de contexto generados cuando se modifica el estado de componentes de interfaz de usuario del conjunto de páginas web de la	<p>Especificación de método de medición: Inicializar la variable #CCM = 0. Por cada página web generada por la aplicación, y a su vez por cada componente de interfaz de usuario, cambiar su estado de la siguiente manera:</p>	<p>▪ Representación: Discreta</p> <p>▪ Tipo de valor: Entero</p> <p>▪ Tipo de escala: Absoluta</p> <p>▪ Unidad: cambios</p>

Cambios de contexto generados cuando se modifica el estado de componentes de interfaz de usuario	aplicación.	<ul style="list-style-type: none"> • En campos de entrada de texto, ingresar algún texto. • En un botón radio o checkbox, hacer un clic en alguna opción. • En una lista desplegable, seleccionar alguna opción. • En secciones de texto en general, hacer un clic en cada párrafo. <p>Si se genera algún cambio de contexto al cambiar el estado de componentes en alguna de las situaciones anteriores, y el mismo no fue informado con anterioridad al usuario, entonces en ese caso $\#CCM = \#CCM + 1$</p> <p>Tipo de método de medición: objetivo</p>	de contexto generados cuando se modifica el estado de componentes
--	-------------	---	---

Atributo 1.2.3. Cambios de contexto generados apropiadamente			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p><i>Nombre:</i> Porcentaje de cambios de contextos apropiados (%CCA)</p> <p><i>Tipo:</i> Indirecta</p> <p><i>Métricas relacionadas:</i></p> <p>1) Número de cambios de contextos apropiados (#CCA)</p> <p>2) Total de cambios de contextos generados (#TCC)</p>	Determinar el porcentaje entre el número de cambios de contextos iniciados apropiadamente y el número total de cambios de contexto generados dentro del conjunto de páginas web de la aplicación.	<p>Especificación de Fórmula:</p> $\%CCA = \left(\frac{\#CCA}{\#TCC} \right) * 100$ <p>Si $\#TCC = 0$ entonces $\%CCA = 100$, ya que no se considera como una situación de incumplimiento sino de inexistencia de cambios de contexto</p>	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %
<p><i>Nombre:</i> Número de cambios de contextos apropiados (#CCA)</p> <p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Cambios de contextos apropiados</p>	Determinar el número de cambios de contextos generados apropiadamente dentro del conjunto de páginas web de la aplicación.	<p>Especificación de método de medición: Inicializar la variable $\#CCA = 0$. Por cada página web generada por la aplicación, pasar el foco entre componentes de interfaz de usuario y modificar su estado. Realizar la operación $\#CCA = \#CCA + 1$ si se cumplen alguna de los siguientes situaciones:</p> <ul style="list-style-type: none"> ▪ Se inicia un cambio de contexto cuando se presiona en un botón o un link (enlace) de la página. ▪ Se inicia un cambio de contexto, ya sea cuando se presiona en un botón o un link de la página ó se 	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: cambios de contextos apropiados

		<p>modifica el estado de un componente pero siempre se informa al usuario del comportamiento de cambio de contexto que se producirá.</p> <p>Tipo de método de medición: objetivo</p>	
--	--	---	--

Sub-Concepto 1.3: Asistencia en el ingreso de datos en formularios

Sub-Concepto 1.3.1: Identificación de errores apropiada

Atributo 1.3.1.1. Uso de notificación textual de error para campos obligatorios			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p><i>Nombre:</i> Porcentaje de notificación textual de error para campos obligatorios (%NTCO)</p> <p><i>Tipo:</i> Indirecta</p> <p><i>Métricas relacionadas:</i> 1) Numero de notificaciones textuales de campos obligatorios (#NTCO) 2) Total de campos obligatorios (#CO)</p>	<p>Determinar el porcentaje entre el número de notificaciones de error para campos obligatorios que no han sido completados y el número total de campos obligatorios, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%NTCO = \frac{\#NTCO}{\#CO} * 100$ <p>Si #CO = 0 entonces %NTCO = 100 ya que no se considera como una situación de incumplimiento sino de inexistencia de campos obligatorios.</p>	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %
<p><i>Nombre:</i> Numero de notificaciones textuales de campos obligatorios (#NTCO)</p> <p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Notificaciones textuales de error para campos obligatorios</p>	<p>Determinar el número de notificaciones textuales de error que se emiten para campos obligatorios que no han sido completados, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de método de medición: Inicializar la variable #NTCO = 0. Por cada página web de la aplicación, y a su vez por cada formulario, dejar todos los campos en blanco y pulsar el botón para enviarlo. Agregar 1 (uno) a #NTCO por cada notificación mediante descripción de texto precisa y en el mismo idioma de la página web en general, que identifica un campo obligatorio que no se completó.</p> <p>En otro caso, agregar 0,5 a #NTCO si existe una notificación textual pero en un idioma diferente al de la página web ó si el mensaje de error se muestra en una página diferente a la del formulario.</p> <p>Agregar 0 (cero) a #NTCO si no se</p>	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Real ▪ Tipo de escala: Absoluta ▪ Unidad: Notificación textual de campo obligatorio

		<p>cumplen las condiciones anteriores. Las notificaciones pueden provenir del lado del cliente y/o del servidor.</p> <p>Tipo de método de medición: subjetivo</p>	
<p><i>Nombre:</i> Total de campos obligatorios (#CO)</p> <p><i>Tipo:</i> Directa</p> <p>Atributo relacionado: Campos obligatorios de un formulario</p>	<p>Determinar el número de campos indicados como obligatorios correspondientes a los formularios del conjunto de páginas web de la aplicación.</p>	<p>Especificación de método de medición: Inicializar la variable #CO = 0. Por cada página web de la aplicación, y por cada formulario, dejar todos los campos en blanco y pulsar el botón para enviarlo. Agregar 1 (uno) a #CO por cada campo que el sistema notifique de alguna manera que debe ser completado, ó por cada campo que el evaluador infiera que es obligatorio.</p> <p>Tipo de método de medición: subjetivo</p>	<p>▪ Representación: Discreta</p> <p>▪ Tipo de valor: Entero</p> <p>▪ Tipo de escala: Absoluta</p> <p>▪ Unidad: Campo obligatorio</p>

Atributo 1.3.1.2. Notificación textual de error apropiada para campos con determinado formato			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p><i>Nombre:</i> Porcentaje de notificación textual de error para campos con formato (%NTCF)</p> <p><i>Tipo:</i> Indirecta</p> <p><i>Métricas relacionadas:</i> 1) Numero de notificaciones textuales de campos con formato (#NTCF) 2) Total de campos con formato (#CF)</p>	<p>Determinar el porcentaje entre el número de notificaciones de error para campos con formato que no han sido completados correctamente y el número total de campos con formato, del conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%NTCF = \frac{\#NTCF}{\#CF} * 100$ <p>Si #CF = 0 entonces %NTCF = 100 ya que no se considera como una situación de incumplimiento sino de inexistencia de campos con formato específico.</p>	<p>▪ Representación: Continua</p> <p>▪ Tipo de valor: Real</p> <p>▪ Tipo de escala: Proporción</p> <p>▪ Unidad: Porcentaje</p> <p>▪ Acrónimo: %</p>
<p><i>Nombre:</i> Numero de notificaciones textuales de campos con formato (#NTCF)</p> <p><i>Tipo:</i> Directa</p> <p>Atributo relacionado: Notificaciones textuales de error para campos con formato</p>	<p>Determinar el número de notificaciones textuales de error que se emiten para campos con un determinado formato y que no han sido completados correctamente, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de método de medición: Inicializar la variable #NTCF = 0. Por cada página web de la aplicación, y a su vez por cada formulario, completar el mismo, ingresando datos erróneos en todos los campos que posean una indicación del formato o valor requerido; o que se infiera que puede requerir un formato particular. Pulsar el botón para enviar el formulario.</p> <p>Verificar si las notificaciones mediante descripción de texto que emite el</p>	<p>▪ Representación: Discreta</p> <p>▪ Tipo de valor: Real</p> <p>▪ Tipo de escala: Absoluta</p> <p>▪ Unidad: Notificación textual de campo con formato</p>

		<p>sistema cumplen con las condiciones siguientes:</p> <ol style="list-style-type: none"> 1) identifica el campo del error 2) proporciona información sobre la entrada inválida 3) indica cómo solucionar el problema <ul style="list-style-type: none"> ▪ Si se cumplen las 3 condiciones anteriores, entonces $\#NTCF = \#NTCF+1$ ▪ Si se cumplen las condiciones 1 y 2, y el idioma del mensaje es el de la página, entonces $\#NTCF = \#NTCF+0,75$ ▪ Si se cumplen las condiciones 1 y 2, y el idioma del mensaje es diferente al de la página, entonces $\#NTCF = \#NTCF+0,50$ ▪ Si se cumple sólo la condición 1, entonces $\#NTCF = \#NTCF+0,50$ ▪ En otro caso, no agregar ningún valor a $\#NTCF$ <p>La notificación puede provenir del lado del cliente y/o del servidor.</p> <p>Tipo de método de medición: objetivo</p>	
<p><i>Nombre:</i> Total de campos con formato ($\#CF$)</p> <p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Campos que requieren un determinado formato</p>	<p>Determinar el número de campos que posean una indicación de un valor o formato requerido correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de método de medición: Inicializar la variable $\#CF=0$. Por cada página web de la aplicación, y a su vez por cada formulario, completar el mismo ingresando datos erróneos en todos los campos que posean una indicación del formato o valor requerido o que se infiera que puede requerir un formato particular.</p> <p>Agregar 1 (uno) a $\#CF$ por cada notificación que emita el sistema, ya sea textual o no, informando que los datos ingresados son incorrectos y que requieren un formato específico.</p> <p>Tipo de método de medición: subjetivo</p>	<p>▪ Representación: Discreta</p> <p>▪ Tipo de valor: Entero</p> <p>▪ Tipo de escala: Absoluta</p> <p>▪ Unidad: Campo con formato específico</p>

Sub-Concepto 1.3.2: Identificación de controles de formularios apropiada

Atributo 1.3.2.1. Representatividad de texto de etiquetas			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p>Nombre: Porcentaje de etiquetas representativas (%ER)</p> <p>Tipo: Indirecta</p> <p>Métricas relacionadas: 1) Numero de etiquetas representativas(#ER) 2) Total de etiquetas(#ET)</p>	<p>Determinar el porcentaje entre el número de etiquetas representativas y el número total de etiquetas, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%ER = \frac{\#ER}{\#ET} * 100$ <p>Si #ET = 0 entonces %ER = 100 ya que no se considera como una situación de incumplimiento sino de inexistencia de etiquetas.</p>	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %
<p>Nombre: Numero de etiquetas representativas (#ER)</p> <p>Tipo: Directa</p> <p>Atributo relacionado: Etiquetas representativas</p>	<p>Determinar el número de etiquetas que se consideran representativas correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de método de medición: Inicializar la variable #ER=0. Por cada página web de la aplicación, y a su vez por cada formulario, observar cada etiqueta correspondiente a uno o varios componentes del formulario.</p> <p>Agregar 1 (uno) a #ER por cada etiqueta diferente, cuyo texto concuerde totalmente con el propósito del componente al cual representa.</p> <p>Tipo de método de medición: subjetivo</p>	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: etiqueta representativa
<p>Nombre: Total de etiquetas de texto (#ET)</p> <p>Tipo: Directa</p> <p>Atributo relacionado: Etiquetas de texto</p>	<p>Determinar el número total de etiquetas de texto de componentes correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de método de medición: inicializar la variable #ET=0. Por cada página web de la aplicación, y a su vez por cada formulario, y agregar 1 (uno) a #ET por cada etiqueta de texto que represente a uno o varios componentes de formulario.</p> <p>Tipo de método de medición: objetivo</p>	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: etiqueta de texto

Atributo 1.3.2.2. Identificación adecuada de controles de formulario con determinado formato			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica

<p>Nombre: Porcentaje de identificación adecuada para campos con formato (%IACF)</p> <p>Tipo: Indirecta</p> <p>Métricas relacionadas: 1) Numero de identificaciones adecuadas para campos con formato (#IACF) 2) Total de campos con formato (#CF - Ver definición dentro de métricas del Atributo 1.3.1.2)</p>	<p>Determinar el porcentaje entre el número de identificaciones adecuadas para campos con formato y el número total de campos con un formato requerido, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%IACF = \frac{\#IACF}{\#CF} * 100$ <p>Si $\#CF = 0$ entonces $\%IACF = 100$, ya que no se considera como una situación de incumplimiento sino de inexistencia de campos con formato.</p>	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %
<p>Nombre: Numero de identificaciones adecuadas para campos con formato (#IACF)</p> <p>Tipo: Directa</p> <p>Atributo relacionado: Indicaciones adecuadas para campos con formato específico</p>	<p>Determinar el número de identificaciones adecuadas para campos que requieran un formato específico correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de método de medición: Inicializar la variable #IACF = 0. Por cada página web de la aplicación, y a su vez por cada formulario, reconocer los controles de formulario que sólo aceptan los datos de entrada del usuario en un formato determinado. Agregar 1 (uno) a #IACF por cada uno de los controles de formulario observados anteriormente que posean una clara identificación e información adecuada sobre el formato esperado.</p> <p>Tipo de método de medición: subjetivo</p>	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: indicaciones adecuadas para campos con formato

Atributo 1.3.2.3. Identificación adecuada de campos obligatorios			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p>Nombre: Porcentaje de identificación adecuada para campos obligatorios (%IACO)</p> <p>Tipo: Indirecta</p> <p>Métricas relacionadas: 1) Numero de identificaciones adecuadas para campos obligatorios (#IACO) 2) Total de campos obligatorios (#CO - Ver</p>	<p>Determinar el porcentaje entre el número de identificaciones adecuadas para campos obligatorios y el número total de campos obligatorios, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%IACO = \frac{\#IACO}{\#CO} * 100$ <p>Si $\#CO = 0$ entonces $\%IACO = 100$, ya que no se considera como una situación de incumplimiento sino de inexistencia de campos obligatorios.</p>	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %

<i>definición dentro de Métricas del Atributo 1.3.1.1)</i>			
<p>Nombre: Numero de identificaciones adecuadas para campos obligatorios (#IACO)</p> <p>Tipo: Directa</p> <p>Atributo relacionado: Identificaciones adecuadas para campos obligatorios</p>	<p>Determinar el número de identificaciones adecuadas para campos obligatorios correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de método de medición: Inicializar la variable #IACO =0. Por cada página web de la aplicación, y a su vez por cada formulario, agregar 1 (uno) a #IACO si se presentan uno o varios de los siguientes casos:</p> <ol style="list-style-type: none"> 1) por cada control de formulario indicado como requerido, cuyo estado se indica como texto dentro del elemento "label" del formulario web, 2) por cada grupo de controles de formularios indicados como requeridos, cuyo texto se incluye en elemento "legend" del formulario web. 3) por cada control o grupo de controles de formulario identificados mediante un símbolo o imagen, y que el significado del indicador se explica antes de su utilización. <p>Nota: en los casos que se requiera, inspeccionar el código fuente de la página web a través de las opciones que brinda cada navegador.</p> <p>Tipo de método de medición: objetivo</p>	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: indicaciones adecuadas para campos obligatorios

Atributo 1.3.2.4. Asociación adecuada de etiquetas con controles de formularios			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p>Nombre: Porcentaje de asociación adecuada entre etiqueta y control de formulario (%AAECF)</p> <p>Tipo: Indirecta</p> <p>Métricas relacionadas:</p> <ol style="list-style-type: none"> 1) Numero de asociaciones adecuadas entre etiqueta y control de formulario (#AAECF) 2) Total de etiquetas de texto (#ET - Ver definición dentro de métricas del Atributo 1.3.2.1) 	<p>Determinar el porcentaje entre el número de asociaciones adecuadas entre etiqueta de texto y control de formulario y el número total de etiquetas de texto, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%AAECF = \frac{\#AAECF}{\#ET} * 100$ <p>Si #ET = 0 entonces %AAECF = 100, ya que no se considera como una situación de incumplimiento sino de inexistencia de etiquetas.</p>	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %
<p>Nombre: Numero de</p>	<p>Determinar el número de</p>	<p>Especificación de método de</p>	<p>▪ Representación:</p>

<p>asociaciones adecuadas entre etiqueta y control de formulario (#AAECF)</p> <p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Asociaciones adecuadas entre etiqueta y control de formulario</p>	<p>asociaciones adecuadas entre etiquetas y controles de formularios, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>medición: inicializar la variable #AAECF =0. Por cada página web de la aplicación, y a su vez por cada formulario, y por cada componente que permita asociación de etiquetas enumerados en la definición del atributo 1.3.2.4, agregar 1 (uno) a #AAECF cuando el texto de la etiqueta que lo representa se incluya dentro del elemento "label" del formulario y el atributo "for" de dicho elemento coincida con el atributo "ID" del control del formulario.</p> <p>Nota: en los casos que se requiera, inspeccionar el código fuente de la página web a través de las opciones que brinda cada navegador.</p> <p>Tipo de método de medición: objetivo</p>	<p>Discreta</p> <ul style="list-style-type: none"> ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: asociaciones adecuadas de etiquetas
---	--	---	---

Atributo 1.3.2.5. Etiquetas de texto de controles de formularios ubicadas adecuadamente			
Datos de Métricas	Objetivo	Método de Cálculo o Medición	Escala Numérica
<p><i>Nombre:</i> Porcentaje de etiquetas ubicadas adecuadamente (%EUA)</p> <p><i>Tipo:</i> Directa</p> <p><i>Métricas relacionadas:</i> 1) Numero de etiquetas ubicadas adecuadamente (#EUA) 2) Total de etiquetas de texto (#ET- Ver definición dentro de métricas del Atributo 1.3.2.1)</p>	<p>Determinar el porcentaje entre el número de etiquetas ubicadas adecuadamente y el número total de etiquetas de texto, correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de Fórmula:</p> $\%EUA = \frac{\#EUA}{\#ET} * 100$ <p>Si #ET = 0 entonces %EUA = 100, ya que no se considera como una situación de incumplimiento sino de inexistencia de etiquetas.</p>	<ul style="list-style-type: none"> ▪ Representación: Continua ▪ Tipo de valor: Real ▪ Tipo de escala: Proporción ▪ Unidad: Porcentaje ▪ Acrónimo: %
<p><i>Nombre:</i> Numero de etiquetas ubicadas adecuadamente (#EUA)</p> <p><i>Tipo:</i> Directa</p> <p><i>Atributo relacionado:</i> Etiquetas ubicadas adecuadamente</p>	<p>Determinar el número de etiquetas ubicadas adecuadamente correspondientes al conjunto de formularios de las páginas web de la aplicación.</p>	<p>Especificación de método de medición: inicializar la variable #EUA =0. Por cada página web de la aplicación, y a su vez por cada formulario, y por cada etiqueta de texto, agregar 1 (uno) a #EUA si se cumplen alguna de las siguientes condiciones:</p> <p>1) si el campo del formulario es una casilla de verificación o botón de opción, la etiqueta debe encontrarse</p>	<ul style="list-style-type: none"> ▪ Representación: Discreta ▪ Tipo de valor: Entero ▪ Tipo de escala: Absoluta ▪ Unidad: etiquetas ubicadas adecuadamente

		<p>inmediatamente después del campo.</p> <p>2) si el campo del formulario no es un botón de opción o casilla verificación, la etiqueta debe estar inmediatamente antes del campo.</p> <p>Tipo de método de medición: objetivo</p>	
--	--	--	--

ANEXO B – DESCRIPCION DE CASOS DE USO

Descripción de Casos de Uso del paquete “Acceso a la WebApp”

CASO DE USO: Iniciar Sesión	
Iniciador	Usuarios: Administrador de Guardias Médicas (AGM de SAF y Área), Supervisor
Precondiciones	Los usuarios deben estar previamente registrados en el Sistema
Flujo básico	
1. Actor: ingresa en la barra de direcciones del navegador, la URL del Sistema de Guardias Médicas	
2. Sistema: despliega en la pantalla la página principal de la WebApp. En el encabezado muestra el nombre de la Aplicación y el Escudo Provincial. En el contenido principal muestra el formulario de Iniciar Sesión para completar los campos CUIL de Usuario y Contraseña	
3. Actor: ingresa N° de CUIL y Contraseña correspondientes y hace un clic en el botón <i>Iniciar Sesión</i>	
4. Sistema: verifica si los datos ingresados son válidos	
5. Sistema: inicia una sesión de usuario y despliega en la pantalla la página web correspondiente con el contenido y el menú apropiado según el tipo de Usuario identificado.	
6. Sistema: muestra debajo del encabezado de la página web principal, un Menú de Usuario donde se muestra el nombre, rol de de usuario identificado, Área o SAF de acceso y un botón con la opción “ <i>Cerrar Sesión</i> ”	
Flujo alternativo 1	Punto 4. Los datos ingresados son inválidos. El sistema muestra un mensaje de alerta, no se autoriza el ingreso del Usuario al Sistema y solicita se revisen los datos ingresados.
Poscondiciones	El sistema autoriza al Usuario a ingresar al Sistema y le brinda la funcionalidad correspondiente de acuerdo a sus permisos

CASO DE USO: Cerrar Sesión	
Iniciador	Usuarios: Administrador de Guardias Médicas (AGM de SAF y Área), Supervisor
Precondiciones	Los actores deben estar ejecutado satisfactoriamente el CU “Iniciar Sesión”
Flujo básico	

1. Actor: selecciona el botón "Cerrar Sesión" dentro del Menú de Usuario que se muestra en la parte superior de la página
2. Sistema: destruye los objetos de la sesión de trabajo actual y finaliza la misma
3. Sistema: notifica al Usuario el cierre de sesión correcto
4. Sistema: redirecciona a la página principal de la WebApp que muestra nuevamente el formulario para Iniciar Sesión
Poscondiciones El sistema finaliza la sesión de un Usuario logueado en el Sistema

Descripción de Casos de Uso del paquete “Administración de Parámetros de la WebApp” para el actor AGM

CASO DE USO: Ver Periodos de Liquidación de Área	
Iniciador	Administrador de Guardias Médicas (AGM) o Supervisor
Precondiciones	Los actores deben haber ejecutado satisfactoriamente el Caso de Uso “Iniciar Sesión”.
Puntos de extensión	Después de que el sistema muestra la lista de Periodos de Liquidación de Área
Flujo básico	
1. Actor:	elige la opción de Menú “Periodos de Liquidación de Área”
2. Sistema:	presenta una página web que contiene una lista de los Periodos de Liquidación de Área, teniendo en cuenta las Áreas a las que tiene acceso el AGM (AGM de Área o SAF) o Supervisor (todas las Áreas).
3. Sistema:	los datos que muestra el sistema para la Lista de Periodos son: N° Liquidación, periodo mensual, Nombre Área, Estado, Fecha Creación, Fecha Liquidación.
4. Sistema:	muestra al pie de la lista, un botón con la opción de “Agregar Periodo de Liquidación Área” y también para cada registro de la lista, presenta un botón con la opción de “Eliminar Periodo de Liquidación”. Si el AGM elige alguna de estas opciones, se extiende la ejecución del presente con el CU “Gestionar Periodos de Liquidación de Área”. Al retomar la ejecución se actualiza la lista de los Periodos de Liquidación de Área de acuerdo a los cambios realizados.

CASO DE USO: Gestionar Periodos de Liquidación de Área	
Iniciador	Administrador de Guardias Médicas (AGM) o Supervisor

Precondiciones	El actor debe haber ejecutado satisfactoriamente el Caso de Uso “Iniciar Sesión” y el Caso de Uso Base “Ver Periodos de Liquidación de Área”.
Flujo básico	
1. Actor: Si el actor eligió en el CU base la opción “Agregar un Periodo Liquidación Área”, continuar en la Sección A , caso contrario, si escogió la opción “Eliminar Periodo Liquidación Área”, continuar en la Sección B .	
2. Sistema: Luego de finalizada cada Sección, el sub-flujo termina y la instancia de caso de uso continúa de acuerdo con el Caso de Uso Base después del punto de extensión.	
Poscondiciones	Se registran o eliminan Periodos de Liquidación de Área que habilitan la Carga de Guardias Médicas para futuras liquidaciones de haberes de un Área en particular.
SECCION A – Agregar un Periodo de Liquidación de Área	
Flujo básico	
1. Sistema: presenta en lugar de la Lista de Periodos, un formulario para ingresar los datos correspondientes: periodo mensual (aaaamm) y un campo de tipo selección que contiene una lista del Nombre de las Áreas a las que tenga acceso el AGM o Supervisor	
2. Actor: ingresa el Periodo mensual.	
3. Actor: selecciona del campo tipo selección, un Área en particular.	
4. Actor: indica que desea guardar el Periodo de Liquidación para el Área.	
5. Sistema: recupera el Id del Área seleccionada por el actor.	
6. Sistema: verifica que no existan registros para el Periodo y Área ingresados.	
7. Sistema: verifica que exista al menos un Contrato de guardia vigente para el Periodo mensual y Área ingresados.	
8. Sistema: asigna al Periodo de Liquidación a registrar, el estado=Abierto y la fecha de creación=fecha actual.	
9. Sistema: registra el Periodo de Liquidación de Área.	
10. Sistema: finaliza la Sección A.	
Flujo alternativo 1	Punto 4. El actor decide cancelar la registración del Periodo de Liquidación. Finaliza la sección A.
Flujo alternativo 2	Punto 6. Existe un registro de Periodo Liquidación de Área para el periodo mensual y Área ingresados. El sistema muestra un mensaje de alerta al actor y solicita se revisen los datos.
Flujo alternativo 3	Punto 7. No existen contratos de guardias vigentes para el periodo mensual y Área ingresados. El sistema no registra el Periodo de Liquidación de Área y muestra un mensaje de alerta al actor.

SECCION B – Eliminar Periodo de Liquidación de Área	
Flujo básico	
1. Sistema: recupera el Id del Periodo de Liquidación de Área seleccionado para Eliminar.	
2. Sistema: verifica que no existan guardias registradas para el Periodo de Liquidación de Área, según lo indicado en RN10	
3. Sistema: pide al actor una confirmación para eliminar el Periodo de Liquidación de Área	
4. Actor: confirma la eliminación del Periodo	
5. Sistema: elimina el Periodo de Liquidación de Área seleccionado	
6. Sistema: Finaliza la sección B	
Flujo alternativo 1	Punto 2. Existen registros de Guardias registradas para el Periodo de Liquidación de Área seleccionado. El sistema informa al actor que no se puede eliminar el Periodo de Liquidación de Área.
Flujo alternativo 2	Punto 4. El usuario decide cancelar la eliminación del Periodo de Liquidación de Área. El sistema no elimina el Periodo de Liquidación de Área.

CASO DE USO: Ver Servicios Internos de Área	
Iniciador	Administrador de Guardias Médicas (AGM) o Supervisor
Precondiciones	Los actores deben haber ejecutado satisfactoriamente el Caso de Uso “Iniciar Sesión”.
Puntos de extensión	Después de que el sistema muestra la lista de Servicios Internos de Área
Flujo básico	
1. Actor: elige la opción de Menú “Servicios Internos de Área”	
2. Sistema: presenta una página web que contiene una lista de los Servicios Internos de Área, teniendo en cuenta las Áreas a las que tiene acceso el AGM (AGM de Área o SAF) o Supervisor (todas las Áreas).	
3. Sistema: los datos que muestra el sistema para la Lista son: Nombre Servicio Interno, Área, Activo (SI/NO)	
4. Sistema: muestra al pie de la lista, un botón con la opción de “Agregar Servicio Interno de Área” y también para cada registro de la lista, presenta un botón con la opción de “modificar” o “eliminar” un registro. Si el AGM elige alguna de estas opciones, se extiende la ejecución del presente con el CU “Gestionar Servicio Interno de Área”. Al retomar la ejecución se actualiza la lista de los Servicios Internos de Área de acuerdo a los cambios realizados.	

CASO DE USO: Gestionar Servicio Interno de Área	
Iniciador	Administrador de Guardias Médicas (AGM) o Supervisor
Precondiciones	El actor debe haber ejecutado satisfactoriamente el Caso de Uso "Iniciar Sesión" y el Caso de Uso Base "Ver Servicios Internos de Área".
Flujo básico	
<p>1. Actor: Si el actor eligió en el CU base la opción "Agregar un Servicio Interno de Área", continuar en la Sección A, si escogió la opción "modificar", continuar en la Sección B, caso contrario, si eligió la opción "eliminar" continuar en la Sección C.</p>	
<p>2. Sistema: Luego de finalizada cada Sección, el sub-flujo termina y la instancia de caso de uso continúa de acuerdo con el Caso de Uso Base después del punto de extensión.</p>	
Poscondiciones	Se registran, modifican o eliminan Servicios Internos de Área en los cuales se realiza una guardia en particular.
SECCION A – Agregar un Servicio Interno de Área	
Flujo básico	
<p>1. Sistema: presenta en lugar de la Lista de Servicios Internos de Área, un formulario para ingresar los datos correspondientes: Nombre, un campo de tipo selección que contiene una lista del Nombre de las Áreas a las que tenga acceso el AGM o Supervisor y una casilla de verificación para indicar si el registro está activo o no.</p>	
<p>2. Actor: ingresa el nombre del Servicio Interno.</p>	
<p>3. Actor: selecciona del campo tipo selección, un Área en particular.</p>	
<p>4. Actor: indica si el registro está activo o desactivo.</p>	
<p>5. Actor: indica que desea guardar el Servicio Interno del Área.</p>	
<p>6. Sistema: verifica que no exista un Servicio Interno con el mismo nombre para el Área seleccionada.</p>	
<p>7. Sistema: registra el Servicio Interno de Área.</p>	
<p>8. Sistema: finaliza la Sección A.</p>	
Flujo alternativo 1	Punto 5. El actor decide cancelar la registración del Servicio Interno de Área. Finaliza la sección A.
Flujo alternativo 2	Punto 6. Existe un registro de Servicio Interno con el mismo nombre para el Área ingresada. El sistema muestra un mensaje de alerta al actor y solicita se revisen los datos.
SECCION B – Modificar Servicio Interno de Área	
Flujo básico	
<p>1. Sistema: presenta un formulario web con los datos del Servicio Interno de Area seleccionado en la sección anterior. Solo se habilitan los campos "nombre" y "activo"</p>	

2. Actor: realiza las modificaciones sobre el registro.	
3. Actor: hace un clic en el botón “Guardar Cambios”	
4. Sistema: verifica que no exista un Servicio Interno con el mismo nombre para el Área seleccionada.	
5. Sistema: guarda los cambios realizados al registro	
6. Sistema: Finaliza la sección B	
Flujo alternativo 1	Punto 3. El actor decide cancelar la registración del Servicio Interno de Área. Finaliza la sección B.
Flujo alternativo 2	Punto 4. Existe un registro de Servicio Interno con el mismo nombre para el Área ingresada. El sistema muestra un mensaje de alerta al actor y solicita se revisen los datos.
SECCION C – Eliminar Servicio Interno de Área	
Flujo básico	
1. Sistema: recupera el Id del Servicio Interno de Área seleccionado para Eliminar.	
2. Sistema: verifica que no existan guardias registradas para el Servicio Interno de Área, según lo indicado en RN10	
3. Sistema: pide al actor una confirmación para eliminar el Servicio Interno de Área	
4. Actor: confirma la eliminación del Servicio Interno	
5. Sistema: elimina el Servicio Interno de Área seleccionado	
6. Sistema: Finaliza la sección C	
Flujo alternativo 1	Punto 2. Existen registros de Guardias registradas para el Servicio Interno de Área seleccionado. El sistema informa al actor que no se puede eliminar el registro.
Flujo alternativo 2	Punto 4. El usuario decide cancelar la eliminación del Servicio Interno de Área. El sistema no elimina el Servicio Interno.

REFERENCIA BIBLIOGRAFICA

- ISO/IEC 9126. (1991). *Information technology - Software product evaluation - Quality characteristics and guidelines for their use.*
- ISO 639-2. (1998). *Codes for the representation of names of languages - Part 2: Alpha-3 code.*
- ISO/IEC 9126-1. (2001). *Software Engineering - Software Product Quality - Part 1: Quality Model.*
- ISO/IEC TR 9126-2. (2003). *Software engineering - Product quality - Part 2: External metrics.*
- BCP 47: *Tags for Identifying Languages.* (09 de 2009). Obtenido de <http://www.rfc-editor.org/rfc/bcp/bcp47.txt>
- ISO/IEC 25010. (2011). *Systems and software engineering. Systems and software product Quality Requirements and Evaluation (SQuaRE). System and software quality models.*
- Modelo–vista–controlador.* (13 de Mayo de 2015). Obtenido de Wikipedia:
<https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
- Aviles Monroy, J. A., Navarro Ramirez, S. T., Toapaxi Acosta, C. V., & Echeverria Briones, P. F. (13 de 05 de 2015). *Tesis de Grado - Sistema de predicción y clasificación en la utilización de recursos humanos para el área de emergencias de un hospital (2006).* Obtenido de <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/3204>
- Becker, P., Lew, P., & Olsina, L. (2012). Specifying Process Views for a Measurement, Evaluation and Improvement Strategy. *Advances in Software Engineering, Software Quality Assurance Methodologies and Techniques*, 1-27.
- Becker, P., Papa, F., & Olsina, L. (2013). Enhancing the Conceptual Framework Capability for a Measurement and Evaluation Strategy. En Q. Sheng, & J. Kjeldskov, *Current Trends in Web Engineering* (págs. 104-116). Springer.
- Boehm, B. W., Brown, J. R., Kaspar, H., & Lipow, M. (1978). *Characteristics of software quality.* Amsterdam: North-Holland Pub. Co.
- Brusilovsky, P., Schwarz, E., & Weber, G. (1996). A Tool for Developing Adaptive Electronic Textbooks on WWW. *World Conference of the Web Society*, (págs. 64-69).

- Bulterman, D., Rutledge, L., Hardman, L., & Van Ossenbruggen, J. (1999). Supporting Adaptive and Adaptable Hypermedia Presentation Semantics. *The 8th IFIP 2.6 Working Conference on Database Semantics (DS-8): Semantic Issues in Multimedia Systems*, (págs. 5-8). Rotorua, New Zealand.
- Ceri, S., Daniel, F., & Matera, M. (2003). Extending WebML for modeling multi-channel context-aware Web Applications. *Web Information Systems Engineering Workshops, 2003. Proceedings. Fourth International Conference on* (págs. 225-233). Roma: IEEE Press.
- Dearle, F. (2010). *Groovy for Domain-Specific Languages*. Packt Publishing.
- Dujmovic, J. (1996). A Method for Evaluation and Selection of Complex Hardware and Software Systems. *The 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG 96 Proceedings*, (págs. 368-378).
- Gallardo, C., & Ahumada, H. (2015). Modelado de un Prototipo de Aplicación Web para la Gestión de Guardias Médicas del Gobierno de Catamarca. *Contribuciones CAIS 2015, 6º Congreso Argentino de Informática y Salud*, (págs. 47-66). Rosario, Santa Fe.
- Gallardo, C., & Quinteros, O. (2014). Análisis del grado de cumplimiento de Pautas de Accesibilidad al Contenido Web a través de la medición y evaluación de Calidad. *II Congreso Argentino de Ingeniería - CADI 2014*. San Miguel de Tucuman, Argentina: Universidad Nacional de Tucumán.
- Gómez, A., Puente, J., Priore, P., Pino, R., & Fernández, I. (2007). Gestión de la plantilla médica en un servicio de urgencias. *XI Congreso de Ingeniería de Organización* (págs. 1357-1364). Madrid:
<http://adingor.es/congresos/web/articulo/detalle/a/747>.
- Hackett, S., Parmanto, B., & Zeng, X. (2004). Accessibility of Internet websites through time. *6th International ACM SIGACCESS Conference on Computers and Accessibility*, (págs. 32-39).
- Judd, C., Faisal Nusairat, J., & Shingler, J. (2008). *Beginning Groovy and Grails: from Novice to Professional*. Apress.
- Kappel, G., Pröll, B., Reich, S., & Retschitzegger, W. (2003). *Web Engineering. The Discipline of Systematic Development of Web Applications*. Germany: John Wiley & Sons.

- Koch, N. (2001). *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process*. Ph.D. thesis. Germany: Ph.D. thesis. Ludwig-Maximilians-University Munich.
- Koch, N., Knapp, A., Zhang, G., & Baumeister, H. (2008). UML-BASED WEB ENGINEERING. An Approach Based on Standards. En G. Rossi, O. Pastor, D. Schwabe, & L. Olsina, *Web Engineering: Modelling and Implementing Web Applications* (págs. 157-191). London: Springer-Verlag.
- McCall, J. A., Richards, P. K., & Walters, G. F. (1977). *Factors in software quality. Volume i. Concepts and definitions of software quality*. Sunnyvale, California: General Electric Co.
- Mich, L., Franch, M., & Gaio, L. (2003). Evaluating and designing Web site quality. *MultiMedia, IEEE Vol 10, Issue 1*, 34–43.
- Murugesan, S. (2008). Web Application development: Challenges and the role of Web Engineering. En G. Rossi, O. Pastor, D. Schwabe, & L. Olsina, *Web Engineering: Modelling and Implementing Web Applications (Human-Computer Interaction Series)* (págs. 7-32). London: Springer.
- Murugesan, S., & Ginige, A. (2005). Web Engineering: Introduction and Perspectives. En W. Suh, *Web Engineering: Principles and Techniques* (págs. 1-30). United States of America: Idea Group.
- Murugesan, S., Deshpande, Y., Hansen, S., & Ginige, A. (1999). Web Engineering: A New Discipline for Development of Web-based Systems. *Proceedings of the First International Conference of Software Engineering (ICSE) Workshop on Web Engineering*, (págs. 1-9). Los Angeles, USA.
- Olsina, L. (1999). *Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web*. Tesis doctoral. La Plata: Universidad Nacional de La Plata.
- Olsina, L., Covella, G., & Rossi, G. (2006). Web Quality. En E. Mendes, & N. Mosley, *Web Engineering* (págs. 109-142). Springer-Verlag.
- Olsina, L., Lew, P., Dieser, A., & Rivera, B. (2012). Updating Quality Models for Evaluating New Generation Web Applications. *Journal of Web Engineering* (págs. 209-246 Vol. 11). Rinton Press, Incorporated .
- Olsina, L., Papa, F., & Molina, H. (2008). How to Measure and Evaluate Web Applications in a Consistent Way. En G. Rossi, o. Pastor , D. Schwabe, & L. Olsina, *Web Engineering: Modelling and Implementing Web Applications* (págs. 385-420). London: Springer-Verlag.

- Pastor, O., Fons, J., Pelechano, V., & Abrahão, S. (2006). Conceptual Modelling of Web Applications: The OOWS Approach. En E. Mendes, & N. Mosley, *Web Engineering* (págs. 277-302). Springer Berlin Heidelberg.
- Pressman, R., & Lowe, D. (2009). *Web Engineering. A practitioner's approach*. New York: McGraw-Hill.
- Rocher, G., & Brown, J. (2009). *The Definitive Guide to Grails. Second Edition*. Apress.
- Schwabe, D., & Rossi, G. (1998). An object oriented approach to Web-based applications design. *Journal Theory and Practice of Object Systems - Special issue objects, databases, and the WWW* (págs. 207-225). New York: John Wiley & Sons.
- Shklar, L., & Rosen, R. (2003). *Web Application Architecture. Principles, protocols and practices*. England: John Wiley & Sons.
- SICATU. (13 de Mayo de 2015). *SICATU*. Obtenido de <https://sites.google.com/site/sicatu30/home>
- Smith, G., & Ledbrook, P. (2009). *Grails in Action*. Greenwich: Manning Publications Co.
- Sullivan, T., & Matson, R. (2000). Barriers to use: usability and content accessibility on the Web's most popular sites. *Proceedings on the 2000 conference on Universal Usability* (págs. 139-144). ACM.
- Tecno Informatica. (13 de 05 de 2015). *GuardiaGes*. Obtenido de <http://www.teinca.com/Proyectos/GuardiasGes/Default.aspx>
- Vigo, M., & Brajnik, G. (2011). Automatic web accessibility metrics: Where we are and where we can go. *Journal Interacting with Computers, vol 23, Issue 2*, 137-155.
- Vigo, M., Arrue, M., Brajnik, G., Lomuscio, R., & Abascal, J. (2007). Quantitative metrics for measuring web accessibility. *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)* (págs. 99-107). ACM.
- WAI, W3C. (05 de Mayo de 1999). *Web Content Accessibility Guidelines 1.0*. Obtenido de <https://www.w3.org/TR/WCAG10/>
- WAI, W3C. (Septiembre de 2005). *Introduction to Web Accessibility*. Obtenido de <http://www.w3.org/WAI/intro/accessibility.php>
- WAI, W3C. (11 de Diciembre de 2008). *Web Content Accessibility Guidelines 2.0*. Obtenido de <https://www.w3.org/TR/WCAG20/>

Web Accessibility Initiative (WAI). (Marzo de 2016). *Web Accessibility Evaluation Tools List*. Recuperado el 20 de Agosto de 2015, de <http://www.w3.org/WAI/ER/tools/>