



Universidad del Azuay

Facultad de Ciencias de la Administración
Escuela de Ingeniería de Sistemas

Sistema informático para kioscos emisores de certificados académicos para estudiantes de la Universidad del Azuay

Tesis Previa a la obtención del título de Ingeniero de Sistemas

Autores:

**Diego Xavier Cedillo Durán
Alfonso José Rivera Donoso**

Directora:

Ing. Janela Encalada

Cuenca - Ecuador
2012

DEDICATORIA

Esta tesis la dedico a mi familia, fuente de soporte, amor y comprensión; que siempre ha sabido estar a mi lado para apoyarme en todo. En especial a mis padres, Humberto y Beatriz, por darme la vida, por apoyarme en mis estudios y en los planes de mi futuro. A mis hermanas, Priscila, por siempre ayudarme y quererme, y Mónica, por siempre tener el coraje de hablar con sinceridad. A mi abuelita Hortensia, por apoyar en mi crianza y consentirme. A mi abuelita Lía, por estar siempre preocupada por toda la familia. Y a mi tía Pia, por ser otra madre más para mí.

Diego Cedillo

Dedico este trabajo a mi familia, que siempre ha sido el apoyo más grande que he tenido durante toda mi vida.

Alfonso Rivera

AGRADECIMIENTOS

Quiero agradecer a todos los profesores que supieron compartir sus conocimientos correctamente en estos 5 años de estudio. En especial, quiero agradecer al Ingeniero Oswaldo Merchán, por siempre escuchar nuestras sugerencias y ayudarnos a formar la idea de la tesis. Quiero agradecer a la Ingeniera Patricia Ortega por ayudarnos con consejos de la tesis. Y, sin duda, mi gran agradecimiento a la Ingeniera Janela Encalada, nuestra directora, que, sin ser nuestra profesora, supo darnos la confianza para poder contar con ella. Finalmente, quiero agradecer a Alfonso Rivera, por ser mi compañero de trabajo y amigo durante esta tesis y durante los casi 5 años de carrera, por su paciencia, apoyo y soporte.

Diego Cedillo

Agradezco a todas las personas que hicieron posible la realización de este trabajo, en especial a la Ingeniera Janela Encalada, al Ingeniero Oswaldo Merchán, a la Ingeniera Patricia Ortega y al final, pero no menos importante a mi amigo Diego.

Alfonso Rivera

ÍNDICE DE CONTENIDOS

<i>Dedicatoria</i>	<i>ii</i>
<i>Agradecimientos</i>	<i>iii</i>
<i>Índice de Contenidos</i>	<i>iv</i>
<i>Índice de Ilustraciones</i>	<i>xii</i>
<i>Resumen</i>	<i>xv</i>
<i>Abstract</i>	<i>xvi</i>
1. Introducción y descripción del proyecto	1
1.1 Introducción	1
1.2 Descripción del Problema	1
1.3 Justificación	2
1.4 Descripción del Objeto de Estudio	2
1.5 Objetivo General	2
1.6 Objetivos Específicos	2
1.7 Resumen del Proyecto y Estructura de la Tesis	3
2. Kioscos Interactivos	5
2.1 Aplicaciones prácticas	6
2.2 Partes de un kiosco	6
2.3 Beneficios	8
2.4 Software de los Kioscos	8
2.4.1 Sistema Operativo	9
2.4.2 Aplicaciones	10
2.4.3 Administración	10
2.5 Seguridad en los Kioscos	11

2.5.1	Puntos Vulnerables	11
2.5.2	Deshabilitando las teclas especiales	13
2.5.3	Pantallas atractivas	14
2.5.4	Otras consideraciones de seguridad	14
2.6	Especificadores del Kiosco requerido para el proyecto.....	15
3.	<i>Interfaces para pantallas táctiles (Touch Screen).....</i>	16
3.1	Características de las interfaces de las pantallas táctiles	16
3.1.1	Tamaño del Objetivo	17
3.1.2	Estrategia para apuntar en una pantalla táctil	18
3.1.3	Teclados Virtuales.....	19
3.1.4	Ángulo de visión y ángulo de operación	20
3.1.5	Estrategia de desplazamiento.....	21
3.1.6	Movimiento de la mano y del brazo	21
3.1.7	Obstrucción de la visión de la pantalla	22
3.1.8	Reacción o retroalimentación	22
3.2	Usos de las pantallas táctiles	23
3.3	Lineamientos para una interfaz táctil.....	23
3.3.1	Reglas principales.....	24
3.3.2	Usabilidad	24
3.3.2.1	Eventos	24
3.3.2.2	Interacción.....	24
3.3.2.3	Operaciones básicas.....	25
3.3.2.4	Cursores, punteros, pantallas de estado	25
3.3.3	Diseño de la Pantalla	25
3.3.4	Gestión del espacio de la pantalla	26
3.3.5	Entrada de Datos	27
3.3.6	Botones y menú	29
3.3.6.1	Botones	29
3.3.6.2	Menús.....	30

3.3.7	Desplazamiento y Navegación	31
3.3.8	Recomendaciones Generales.....	32
4.	Web Services.....	34
4.1	Definición.....	34
4.1.1	Agentes y Servicios.....	35
4.1.2	Solicitantes y proveedores.....	35
4.1.3	Descripción del Servicio.....	35
4.1.4	Semántica	36
4.1.5	Conexión con un web service	36
4.2	Necesidad de web services	37
4.3	Arquitectura Orientada a Servicios (SOA)	38
4.3.1	Sistemas Distribuidos.....	38
4.3.2	Servicios Web y estilos Arquitectónicos	40
4.3.3	Arquitecturas REST	41
4.4	Tecnologías Web Services.....	42
4.4.1	XML.....	43
4.4.2	SOAP	44
4.4.3	WSDL.....	46
4.5	Utilizando Web Services	50
4.6	Seguridad en Web services	52
4.6.1	Políticas de seguridad.....	53
4.6.2	Amenazas de seguridad a nivel de mensajes	54
4.6.3	Requerimientos de seguridad	55
4.6.3.1	Mecanismos de Autenticación	56
4.6.3.2	Autorización	56
4.6.3.3	Integridad y confidencialidad de los datos	56
4.6.3.4	Integridad de las transacciones y comunicaciones	56
4.6.3.5	No Repudio	57
4.6.3.6	Integridad y confidencialidad de mensajes emisor-receptor.....	57

4.6.3.7	Pistas de auditoría	57
4.6.3.8	Ejecución distribuida de políticas de seguridad	57
4.6.4	Consideraciones de seguridad de la arquitectura	57
4.6.5	Desafíos a la seguridad de los servicios Web	59
5.	<i>Especificación de Requisitos de Software (ERS)</i>	62
5.1	Introducción	62
5.1.1	Propósito	62
5.1.2	Ámbito del Sistema	62
5.1.3	Definiciones, Acrónimos y Abreviaturas	63
5.1.4	Referencias	63
5.1.5	Historial de Versiones	63
5.2	Descripción general	64
5.2.1	Perspectiva del Producto	64
5.2.1.1	Dependencias del producto.	64
5.2.1.2	Interfaces hardware	64
5.2.1.3	Interfaces de Software	65
5.2.2	Funciones del Producto	65
5.2.3	Características de los usuarios	65
5.2.4	Restricciones.....	66
5.2.5	Suposiciones y Dependencias	66
5.2.6	Requisitos Futuros	66
5.3	Requisitos Específicos	67
5.3.1	Requisitos Funcionales	67
5.3.1.1	Descripción de Actores	67
5.3.1.2	Descripción de Casos de uso	69
5.3.2	Requerimientos No funcionales	90
5.3.2.1	Funcionalidad	90
5.3.2.2	Confiabilidad	91
5.3.2.3	Usabilidad	91

5.3.2.4	Eficiencia	92
5.3.2.5	Mantenibilidad	92
5.3.2.6	Portabilidad	93
5.4	Interfaces Externas	94
5.4.1	Interfaces del Software	94
5.4.2	Interfaces de usuario.....	95
5.4.3	Interfaces de Comunicaciones.....	95
6.	<i>Diseño del sistema propuesto</i>.....	96
6.1	Diseño del sistema mediante UML.....	96
6.1.1	Conceptos de Orientación a Objetos	97
6.1.2	Análisis y Diseño Orientado a Objetos	97
6.1.3	Arquitectura UML	98
6.2	Modelado UML.....	99
6.2.1	Modelado Estructural	99
6.2.1.1	Diagrama de Clases.....	100
6.2.1.2	Diagrama de Paquetes	105
6.2.1.3	Diagrama de Despliegue.....	107
6.2.2	Modelado de Comportamiento.....	109
6.2.2.1	Diagrama de Casos de Uso	109
6.2.2.2	Diagrama de Actividad	110
6.2.2.3	Diagramas de Interacción – Los Diagramas de Secuencia.....	111
6.2.2.4	Diagrama de Estados.....	116
6.2.3	Modelado Arquitectónico.....	118
6.3	Diseño de la Interfaz gráfica del Kiosco	121
6.3.1	Pantalla Inicial	121
6.3.2	Inicio de Sesión.....	122
6.3.3	Menú de Selección de Certificados	125
6.3.4	Vista Previa del Certificado	126
6.3.5	Diseño general de la GUI.....	128

7.	Desarrollo y Pruebas	131
7.1	Problemas encontrados y soluciones	131
7.1.1	Resolución de Pantalla	131
7.1.1.1	Descripción del Problema	131
7.1.1.2	Alternativas	131
7.1.1.3	Solución	132
7.1.2	Teclado virtual	132
7.1.2.1	Descripción	132
7.1.2.2	Alternativas	132
7.1.2.3	Solución	132
7.1.3	Plataforma de Reportes	133
7.1.3.1	Descripción del Problema	133
7.1.3.2	Alternativas	133
7.1.3.3	Solución	133
7.1.4	Vista previa del certificado	134
7.1.4.1	Descripción del problema.....	134
7.1.4.2	Alternativas	134
7.1.4.3	Solución	134
7.1.5	Lenta Apertura de la vista previa del certificado	135
7.1.5.1	Descripción del Problema	135
7.1.5.2	Alternativas	135
7.1.5.3	Solución	135
7.2	La plataforma de reportes	135
7.2.1	Elección de la plataforma para los reportes.....	135
7.2.2	Uso de la plataforma BIRT.....	137
7.3	Manual de creación de nuevos certificados	139
7.3.1	Creación de servicios web	139
7.3.2	Creación del nuevo certificado.....	140
7.3.2.1	Nuevo Reporte Tipo Certificado.....	140
7.3.2.2	Nueva Fuente de Datos	142

7.3.2.3	Tamaño de Hoja.....	147
7.3.2.4	Inclusión de nuevos datos en el certificado	147
7.3.3	Integración del certificado al sistema.....	151
7.4	Plan de Pruebas	152
7.4.1	Pruebas de Servicios Web.....	152
7.4.1.1	Web Service consultarDatosEstudiante.....	152
7.4.1.2	Web Service consultarMatricula	153
7.4.1.3	Web Service verificarMatricula	154
7.4.1.4	Web Service consultarNotasActuales.....	154
7.4.1.5	Web Service consultarNotasHistoricas.....	155
7.4.1.6	Web Service consultarPensum	156
7.4.1.7	Web Service consultarHorario	157
7.4.1.8	Web Service verificarEgresado.....	158
7.4.1.9	Web Service consultarDatosEgresado	158
7.4.2	Pruebas de Sesión, interfaz gráfica y de certificados.....	159
7.4.2.1	Ingreso al sistema	159
7.4.2.2	Selección del Certificado de Notas Actuales	161
7.4.2.3	Selección del Certificado de Horario.....	162
7.4.2.4	Selección del Certificado de Matricula.....	162
7.4.2.5	Selección del Certificado de Notas Históricas	163
7.4.2.6	Selección del Certificado de Pensum	163
7.4.2.7	Selección del Certificado de Egresado	164
8.	Conclusiones y Recomendaciones.....	165
8.1	Conclusiones.....	165
8.2	Recomendaciones	167
10.	Referencias.....	170
11.	Anexos	173
11.1	ANEXO 1: CERTIFICADO DE MATRICULA.....	173

11.2	ANEXO 2: CERTIFICADO DE EGRESADO	174
11.3	ANEXO 3: CERTIFICADO DE NOTAS HISTÓRICAS	175
11.4	ANEXO 4: CERTIFICADO CON EL PENSUM	177
11.5	ANEXO 5: CERTIFICADO CON EL HORARIO	179
11.6	ANEXO 6: CERTIFICADO DE NOTAS DEL CICLO ACTUAL	180

ÍNDICE DE ILUSTRACIONES

Figura 2-1: Ejemplo de un kiosco interactivo.....	5
Figura 2-2: Partes de un Kiosco ¹	7
Figura 2-3: Comparación entre un Sistema Operativo normal (izquierda) y uno exclusivo para kioscos.....	9
Figura 2-4: El ciclo de vida del software de un kiosco.....	12
Figura 2-5: Comparación entre una aplicación vulnerable y una protegida.	13
Figura 2-6: Ejemplo de Teclado virtual	14
Figura 3-1: Ejemplo de Interfaz en forma de asistente.....	16
Figura 3-2: Ángulo entre la pantalla y la mesa	20
Figura 3-3: Operaciones básicas en un touch screen, organizadas de la mejor a la peor. ..	25
Figura 3-4: Ingreso de datos mediante incremento y decremento de valores.	27
Figura 3-5: Ejemplos de teclados virtuales: teclado numérico, teclado telefónico, y alfanumérico.....	28
Figura 3-6: Ejemplos de menús con botones. A la izquierda con apariencia de botones y diferenciando los botones y menús mediante formas y estilos. A la derecha con imágenes y ocultando la apariencia de botones.	30
Figura 3-7: Botones de desplazamiento hacia arriba (izquierda) y hacia abajo (derecha). .	32
Figura 4-1: El proceso de conexión con un web service.	36
Figura 4-2: Pila de la Arquitectura Web Service.....	42
Figura 4-3: Ejemplo XML Schema.....	44
Figura 4-4: Ejemplo de invocación de servicio mediante SOAP.	45
Figura 4-5: Ejemplo de respuesta de servicio mediante SOAP. ¹³	45
Figura 4-6: Ejemplo de WSDL para la operación GetLastTradePrice del servicio StockQuoteService.	49
Figura 4-7: Comparación y analogías entre WSDL 1.1 y 2.0	50
Figura 4-8: Seguridad End-to-End.....	54
Figura 5-1: Diagrama de Casos de uso esencial.....	65
Figura 5-2: Diagrama de Casos de uso extendido	68

Figura 5-3: Diagrama de Actividad	69
Figura 6-1: Diagrama de Clases de los web services.	102
Figura 6-2: Diagrama de Clases de la generación de certificado.....	104
Figura 6-3: Diagrama de Paquetes.....	106
Figura 6-4: Diagrama de Despliegue	108
Figura 6-5: Diagrama de Secuencia General.	113
Figura 6-6: Diagrama de Secuencia para Identificar Estudiante e indicar los certificados disponibles.	114
Figura 6-7: Diagrama de Secuencia para generar certificados y visualizarlos.	115
Figura 6-8: Diagrama de Secuencia de la impresión y generación de la deuda del certificado.....	116
Figura 6-9: Diagrama de Estados del Kiosco emisor de certificados	117
Figura 6-10: Diseño arquitectónico.....	118
Figura 6-11: Diseño arquitectónico de certificados, diagrama inicial.	119
Figura 6-12: Diseño arquitectónico de Certificados, con la plataforma BIRT.....	119
Figura 6-13: Diseño arquitectónico específico de Sistema de Kioscos emisores de certificados.....	120
Figura 6-14: Pantalla Inicial o de Bienvenida	121
Figura 6-15: Pantalla de Inicio de Sesión.....	122
Figura 6-16: Teclado telefónico.	123
Figura 6-17: Pantalla de inicio de sesión al ingresar la contraseña.	123
Figura 6-18: Teclado Virtual con distribución QWERTY en minúsculas.	124
Figura 6-19: Teclado Virtual con distribución QWERTY en mayúsculas.	124
Figura 6-20: Teclado Virtual de símbolos.....	125
Figura 6-21: Pantalla con el menú de selección de certificados disponibles.	125
Figura 6-22: Pantalla Vista previa del certificado.	127
Figura 6-23: Pregunta de confirmación para la impresión del certificado.	127
Figura 6-24: Experiencia del Usuario. Flujo de Interacción Usuario – Aplicación.....	129
Figura 7-1: Esquema de trabajo de JasperReports.....	136
Figura 7-2: Esquema de Trabajo de BIRT	138

Figura 7-3: Componentes utilizados por la GUI del Kiosco.....	139
Figura 7-4: Asistente de creación de un nuevo certificado.....	140
Figura 7-5: Elementos incluidos en la plantilla Certificado.	141
Figura 7-6: Vista previa de la plantilla certificado.	141
Figura 7-7: Nuevo Data Set	142
Figura 7-8: Asistente para crear nuevos Data Set.....	143
Figura 7-9: Seleccionar el servicio web dentro del WSDL.	143
Figura 7-10: SOAP Request para consultarPensum.....	144
Figura 7-11: Configuración de Parámetros.....	145
Figura 7-12: Selección de la expresión de consulta XPath.	145
Figura 7-13: Selección y configuración de elementos a formar parte del Data set.....	146
Figura 7-14: Crear un enlace entre el parámetro del reporte y el parámetro del data set. .	146
Figura 7-15: Selección del tamaño de hoja en el certificado.....	147
Figura 7-16: Tabla del Data set Pensum	147
Figura 7-17: Insertar nueva forma de agrupar datos.....	148
Figura 7-18: Creación de un nuevo grupo.	148
Figura 7-19: Filtros para los datos mostrados en el grupo.....	149
Figura 7-20: Diseño del certificado del pensum académico del estudiante.....	150
Figura 7-21: Inclusión del nuevo certificado en el directorio.....	151
Figura 7-22: Error: usuario no habilitado	160
Figura 7-23: Error: Estudiante no matriculado o no está egresado.....	160
Figura 7-24: Error: Usuario o contraseña no coinciden con un estudiante.....	160
Figura 7-25: Error: contraseña de menor longitud a la permitida.....	161
Figura 7-26: Error: no se proporciona contraseña.....	161

RESUMEN

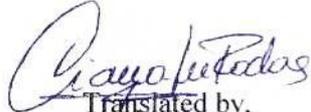
La presente tesis describe los fundamentos teóricos y prácticos para la construcción de un sistema informático para kioscos emisores de certificados académicos, para uso de los estudiantes de la Universidad del Azuay. Se propone desarrollar una interfaz gráfica táctil para los kioscos, servicios web que obtengan la información necesaria, y la correspondiente validación y verificación del sistema desarrollado. Para lograr esto, se hace uso de los conceptos sobre kioscos interactivos, los lineamientos y recomendaciones para el diseño de interfaces gráficas táctiles, la arquitectura SOA, la tecnología EAI, los conceptos de análisis y diseño orientado a objetos, y el lenguaje UML.

ABSTRACT

The present thesis describes the theoretical and practical fundamentals for the construction of an informatics system for stands that issue academic certificates for students of the University of Azuay. The development of a tactile graphic interface for the stands is proposed, as well as web services that obtain information, and the corresponding validation and verification of the system. In order to do this, the concepts regarding interactive stands, the guidelines and recommendations for the design of tactile graphic interfaces, SOA architecture, EAI technology, the concepts of analysis and object design, and UML language are employed.



UNIVERSIDAD DEL
AZUAY
DPTO. IDIOMAS



Translated by,
Diana Lee Rodas

CAPÍTULO I

1. INTRODUCCIÓN Y DESCRIPCIÓN DEL PROYECTO

1.1 Introducción

Un sistema informático es creado con el propósito de automatizar un proceso existente o generar uno mejorado. Lo que permite: agilizar trámites, reducir la carga de trabajo y facilitar la vida de las personas en general. Es por ello que, en el sistema propuesto se plantea automatizar la manera en que los estudiantes obtienen sus certificados académicos por parte de la Universidad del Azuay.

El sistema planteado no constituye únicamente una automatización al proceso existente. En realidad, propone un ligero cambio al modo en que los estudiantes solicitan sus certificados y de cómo los reciben. El nuevo sistema plantea que los certificados sean solicitados y emitidos por medio de un Kiosco Interactivo en el cual un estudiante se identifica por su Usuario y Contraseña, solicita el certificado que desea y el sistema lo imprime automáticamente. Este nuevo proceso permite brindar un mejor servicio a los estudiantes de la Universidad y aligerar la carga de trabajo de los empleados.

1.2 Descripción del Problema

En la Universidad del Azuay se ha podido observar que existen demoras injustificadas en el momento de otorgar certificados a los estudiantes, teniendo que esperar un día para obtener el certificado pedido. El proceso del sistema actual comienza cuando un estudiante se acerca a la solicitar un certificado, las secretarías de la Facultad lo redactan y se espera hasta que el secretario lo firme, siendo estas dos últimas fases las causantes de la demora; problema que podría ser resuelto con un sistema que automatice el procedimiento descrito en el *Manual*

de Procesos de Certificaciones Estándar, el cual es el resultado de una consultoría realizada a la Universidad del Azuay por “Dynamix Solutions Consultores Integrales”.

1.3 Justificación

Los estudiantes son parte importante de la Universidad, y por tanto, se les debe brindar el mejor servicio en el menor tiempo posible. En la actualidad, para el caso de emisión de certificados, esta premisa no se cumple; pues, se ha observado que se requiere múltiples visitas al campus para finalmente obtener el certificado. Por esto, resulta necesario implementar un punto de atención automático para emitir los certificados que los alumnos requieran, brindándoles una atención inmediata.

1.4 Descripción del Objeto de Estudio

El sistema podrá ser implementado en un kiosco interactivo con una impresora, y permitirá receptor las peticiones de certificados, recargar su costo a la matrícula y entregar los certificados inmediatamente, sin que pierdan su validez y legalidad, brindando las mayores facilidades a los alumnos de la Universidad.

1.5 Objetivo General

- Desarrollar un sistema informático para kioscos emisores de certificados académicos para estudiantes de la Universidad del Azuay.

1.6 Objetivos Específicos

- Desarrollar un sistema para interfaz táctil, que permita emitir certificados académicos.
- Desarrollar los procedimientos almacenados necesarios para que sean utilizados por el sistema propuesto.

- Validar el software desarrollado para brindar el servicio propuesto.

1.7 Resumen del Proyecto y Estructura de la Tesis

La presente tesis tiene como fin dotar a la Universidad del Azuay un sistema interactivo para kioscos con pantallas táctiles, que le permita mejorar los procesos internos y brindar a los estudiantes un mejor servicio, permitiéndoles la obtención automática e inmediata de certificados académicos.

Para el desarrollo del sistema propuesto, se pretende maximizar la portabilidad creando servicios web. Los mismos que permitan la obtención de los datos necesarios para la emisión de los certificados, y sirvan de base para la creación de futuras aplicaciones. Además, el sistema debe ser capaz de aceptar nuevos tipos de certificados sin mayor esfuerzo, para así, aumentar el número de certificados disponibles de manera automática para los estudiantes

Para lograr este fin, se ha realizado una investigación sobre los kioscos interactivos, plasmada en el Capítulo II “Kioscos Interactivo”. En este capítulo se abordan los conceptos sobre kioscos interactivos, sus usos, partes, beneficios, vulnerabilidades y las recomendaciones que deben tomarse en cuenta en aplicaciones que ejecutan.

Debido a que la aplicación propuesta debe ser ejecutada con una interfaz de pantalla táctil, es necesario conocer las características de estas interfaces. Más aún, es necesario tomar en cuenta los lineamientos y recomendaciones al momento de diseñar interfaces gráficas para estos dispositivos. Estas características, lineamientos y recomendaciones se encuentran en el Capítulo III “Interfaces para pantallas táctiles (Touch Screen)”.

Como el sistema debe contar con una arquitectura que incluye a servicios web, se realiza una definición de los mismos en el Capítulo IV “Web Services”. Este

capítulo, además, da una visión sobre la arquitectura orientada a servicios, las aplicaciones y usos de los servicios web, las tecnologías que se utilizan y las consideraciones de seguridad que se deben tomar en cuenta al utilizarlos.

Como parte del análisis realizado para la construcción del sistema propuesto, se desarrolló la Especificación de Requisitos de Software (ERS) del Capítulo V. En ella, se describen los requisitos específicos funcionales, agrupados por medio de casos de uso; y los requerimientos no funcionales, mediante el estándar ISO/IEC 9126. En la descripción de cada caso de uso se indican, entre otras cosas, los requisitos asociados y el flujo de eventos del caso de uso, basados en el diagrama de actividad.

En base al análisis realizado, se desarrolla el “Diseño del sistema propuesto” del Capítulo VI. En este capítulo, se da una introducción al análisis y diseño orientado a objetos mediante diagramas UML. Con ello, se presenta el diseño del sistema mediante el modelado estructural, el modelado de comportamiento y el diseño arquitectónico. Por último, en base a los conceptos del Capítulo III, se presenta el Diseño de la interfaz gráfica del usuario para el kiosco interactivo con pantallas táctiles.

En base a los conceptos investigados y al análisis y diseño realizado, se realizó el desarrollo de sistema. Las decisiones tomadas en este proceso y sus razones, fueron documentadas en el Capítulo VII “Desarrollo y Pruebas”. En este capítulo, además, se presenta un manual para la creación de nuevos certificados académicos y el plan de pruebas ejecutado en la verificación del sistema.

Finalmente, en el Capítulo VIII, se presentan las conclusiones de esta tesis y las recomendaciones que se hacen a la Universidad del Azuay sobre el futuro de este sistema y las partes que lo conforman.

CAPÍTULO II

2. KIOSCOS INTERACTIVOS

Los Kioscos son terminales computarizadas, que nos ayudan a realizar trabajos específicos (Figura 2-1). Se los encuentra, en su mayoría, en sitios públicos, como por ejemplo: hospitales, empresas públicas, aeropuertos, universidades, empresas privadas, etc.



Figura 2-1: Ejemplo de un kiosk interactivo

Como una computadora, un kiosk puede poseer: un acceso a internet, capacidad de reproducción de archivos multimedia, envío de correo electrónico o el acceso a aplicaciones según sea el objetivo de este. En cambio, a diferencia de las computadoras, los kioscos realizan tareas específicas de una manera fácil y rápida, para ello, tienen una interfaz simple que cualquier usuario, sin entrenamiento, la pueda utilizar.

2.1 Aplicaciones prácticas

Existen muchos tipos de aplicaciones prácticas para los kioscos, pero las principales o más utilizadas son:

- **Sistemas de Check-In de aerolíneas:** Este tipo de Kioscos son de gran utilidad, pues permiten, a las personas que han realizado su Check-In por Internet, emitir su pase de abordaje. Para ello, el Kiosco incluye una pequeña impresora de boletos. Estos kioscos resultan especialmente útiles para personas que no poseen equipaje.
- **Fotográficos:** Se usan para la impresión de fotografías, reduciendo el tiempo de espera del usuario y optimizando el proceso de impresión. Permitiendo al usuario seleccionar, por sí mismo, las fotos que desea imprimir.
- **Servicios de Internet:** Estos kioscos se utilizan principalmente en áreas públicas y proveen la facilidad de navegar por la web o enviar correo electrónico.
- **Información para trabajadores:** Su principal utilidad está en mostrar a los trabajadores de las empresas información acerca de los productos que venden, para así, dar un mejor servicio a sus clientes.
- **Información al público:** Estos kioscos son los más comunes y permiten brindar información específica a los usuarios en general. Se usan comúnmente en empresas públicas o privadas en la parte de servicio al cliente.

2.2 Partes de un kiosco

Los kioscos vienen en muchas formas y tamaños y, normalmente, se arman dependiendo de las necesidades que tenga la empresa y el servicio que se va a

prestar. Esto se debe a que los componentes que pueden poseer son variados, siendo los más comunes son los mostrados en la Figura 2-2¹:

1. La cabina: es la parte estética del kiosco y dentro de esta se encontrarán los componentes del mismo.
2. Chasis o case: es una de las partes más importantes del hardware del kiosco. Este incluirá el mainboard, procesador, memoria, disco duro y los puertos de conexión externa. Generalmente, en un kiosco, el chasis se encuentra unido a la pantalla.
3. Pantalla: en ella se mostrarán la información del kiosco, estas pueden ser pantallas comunes o táctiles, según sea el caso. Los tamaños más comunes varían entre 15" y 19".

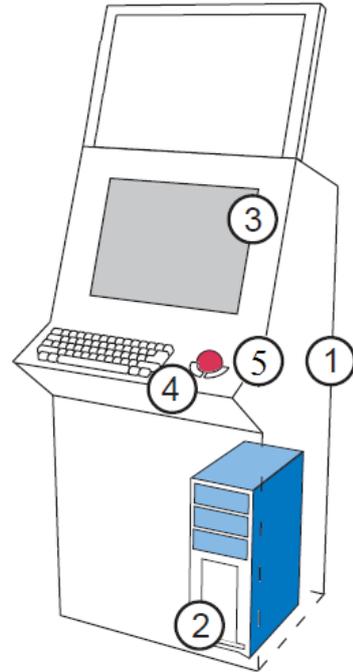


Figura 2-2: Partes de un Kiosco¹

4. Teclado: su función será de introducir la información al kiosco. Los teclados son necesarios en casos excepcionales.
5. Ratón: este servirá para desplazarse en la pantalla, como en una computadora normal. Al igual que el teclado, sólo se lo utiliza en casos excepcionales.

Los kioscos también pueden tener otros periféricos como: impresoras, parlantes, lector de código de barras, lector de huella digital, cámara, etc. según sea necesario.

¹ (Wirespring)

2.3 Beneficios

Los beneficios que se pueden mencionar en cuanto al uso de kioscos son los siguientes:

- Reducen la carga de trabajo: pueden realizar tareas, que al personal le tomaba mucho tiempo, en pocos segundos; ayudando al personal a dedicarse a otras tareas más importantes.
- Confiabilidad: Los kioscos están diseñados para estar funcionando todo el tiempo sin caerse o sin ser reseteados. Esto hace que sean confiables y funcionen sin inconvenientes.
- Seguridad: Los kioscos proveen seguridad a los procesos porque los automatizan, eliminando de esta manera los errores que pueden existir por la intervención humana.
- Locación: Los kioscos pueden estar en cualquier sitio según sea la necesidad del cliente.
- Estadísticas: Los kioscos pueden dar estadísticas de uso, para así permitir tomar decisiones importantes a la empresa en los ámbitos que sean requeridos.

2.4 Software de los Kioscos

El software en los kioscos se divide en tres partes muy importantes:

- Sistema Operativo.
- Aplicaciones.
- Administración.

Adicionalmente, herramientas especiales de desarrollo pueden ser usadas para la creación de aplicaciones que funcionen en los kioscos interactivos.

2.4.1 Sistema Operativo

Como en una PC, los kioscos interactivos tienen un sistema operativo, el cual, constituye un conjunto de herramientas que le permiten al kiosco identificar el hardware y ejecutar los programas. La Figura 2-3 permite ilustrar algunas de estas herramientas.

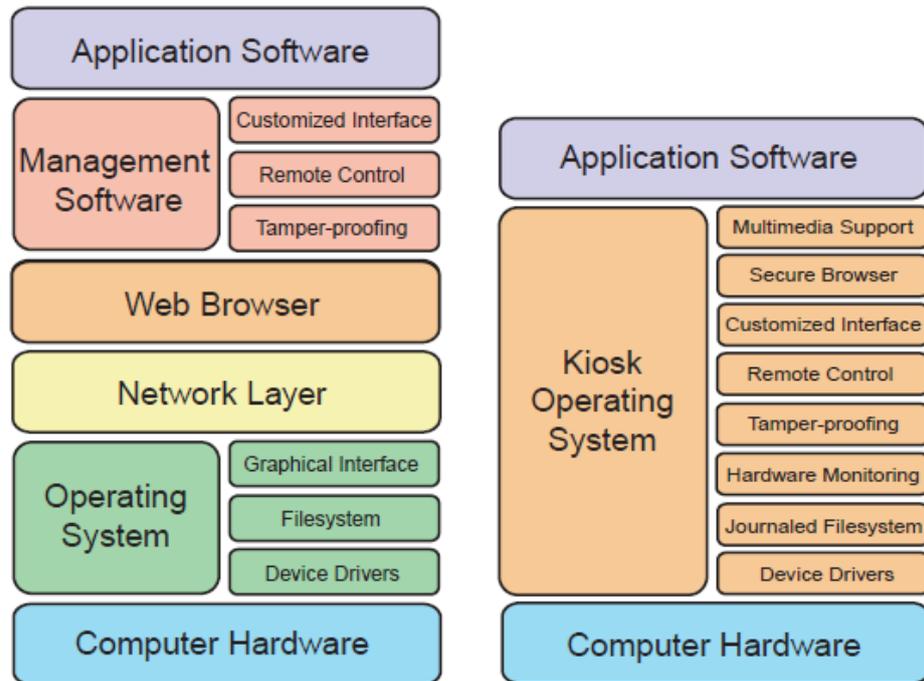


Figura 2-3: Comparación entre un Sistema Operativo normal (izquierda) y uno exclusivo para kioscos²

En la mayoría de casos, los kioscos utilizan una versión modificada del sistema operativo Microsoft Windows (Figura 2-3 Izquierda), porque es mucho más sencillo de manejar. Sin embargo, hoy en día existen otras opciones como por ejemplo, WireSpring's FireCast OS: un sistema operativo específicamente creado para kioscos (Figura 2-3 derecha).

² (Wirespring)

2.4.2 Aplicaciones

Las aplicaciones en los kioscos son variadas, y personalizadas de acuerdo a la funcionalidad que se le quiera dar al kiosco. En algunos casos, los dueños de los kioscos hacen modificaciones en los programas ya existentes, para hacerlos más aceptables para el manejo del público en general. Algunos kioscos usan un ambiente web con un navegador modificado, con botones especiales e imágenes mucho más sofisticadas para sus clientes.

Las aplicaciones son la clave para explotar al máximo las funcionalidades del kiosco. Cabe destacar que las aplicaciones deben ser pensadas en maximizar la facilidad de uso, limitando el número de acciones que debe realizar el usuario. También se debe tomar en cuenta que, en la mayoría de los kioscos, la interacción es mediante una pantalla táctil, por lo que, los botones de la aplicación deben poseer el tamaño adecuado.

2.4.3 Administración

La administración de kioscos usa las ventajas de la administración remota, ya que, permite bajar los costos de mantenimiento y mantener los contenidos mostrados, todos los días. Los kioscos usan un software especial para conectarse a servidores externos (comúnmente vía internet) y, de esta manera, ser controlados remotamente. El software de administración de kioscos es esencial para su eficiente funcionamiento en red y para verificar su correcto funcionamiento. Además, este software es responsable de tres tipos de tareas:

- Seguridad.
- Tareas de la aplicación.
- Administración de la Aplicación.

2.5 Seguridad en los Kioscos

(Basado en el artículo publicado en KioskMarketplace de (Slawsky))

El peor inconveniente o problema de seguridad que se puede dar en un kiosco, es el acceso de los hackers al sistema operativo, causando estragos en el kiosco y accediendo a la red a la que se encuentra conectado. De igual manera, es peligroso el acceso de un usuario al sistema de ficheros donde puede existir información sensible.

Los hackers, generalmente, atacan a un kiosco simplemente por el reto que representa. Sin embargo, la mayor recompensa que pueden obtener de su trabajo es la información que se encuentra en el kiosco y la posibilidad de acceder a la red de la empresa a la que el kiosco se encuentra conectado.

Para asegurar los kioscos y evitar robos, se los encierran dentro de la cabina. No obstante, el componente menos visible, el sistema operativo, puede ser la causa de la mayor vulnerabilidad. Por ejemplo, un usuario frustrado puede manipular la interfaz violentamente presionando un botón o una combinación de ellos que le permita el acceso al sistema.

2.5.1 Puntos Vulnerables

Para evitar inconvenientes, el kiosco siempre debe estar protegido y mantenerse en la aplicación que le otorga su funcionalidad. Como lo dice Stephanie Kropkowski: "Si el sistema de un kiosco no está siendo usado, entonces sí, es fácil acceder a cualquier parte del kiosco."³

Un kiosco, generalmente, tiene dos tipos de software: la aplicación que corre y el software del sistema. El primero da la funcionalidad al kiosco pero, a su vez, crea la vulnerabilidad. Mientras el segundo, la protege al kiosco de la misma. Este software

³ (Slawsky)

debe permitir bloquear los componentes que se requieran, para evitar accesos a otras partes que no correspondan a los propósitos del kiosk.

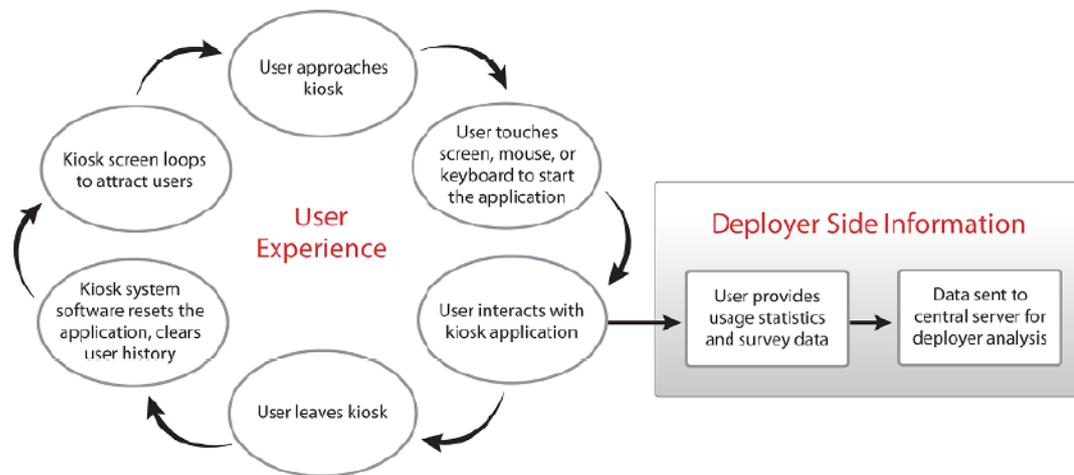


Figura 2-4: El ciclo de vida del software de un kiosk.⁴

Para que un sistema de un kiosk sea seguro, sabemos que debe encontrarse siempre dentro de la aplicación que le brinda su funcionalidad. Para ello, se recomienda que el ciclo de vida del kiosk sea el mostrado en la Figura 2-4. Primero, el kiosk se encuentra mostrando la pantalla principal, que atrae a los usuarios. Cuando un usuario se aproxima y toca la pantalla, la aplicación se inicia. A continuación, el usuario interactúa con la aplicación, lo que permite recolectar información y estadísticas de uso. En el momento que el usuario deja el kiosk, el sistema resetea la aplicación, cierra sesión y borra el historial. Regresando a la pantalla principal.

Otro ejemplo sobre la permanencia del kiosk en la aplicación funcional, se muestra en la Figura 2-5, la cual, indica la diferencia entre: una aplicación que expone el kiosk a ataques (izquierda), de aquella que lo protege (derecha). La primera, deja libre la barra de inicio de Windows y la de herramientas del explorador, para que puedan ser manipuladas por el usuario. En el segundo caso,

⁴ Tomado de (Slawsky)

la aplicación cubre estas vulnerabilidades y utiliza una barra de herramientas propia.

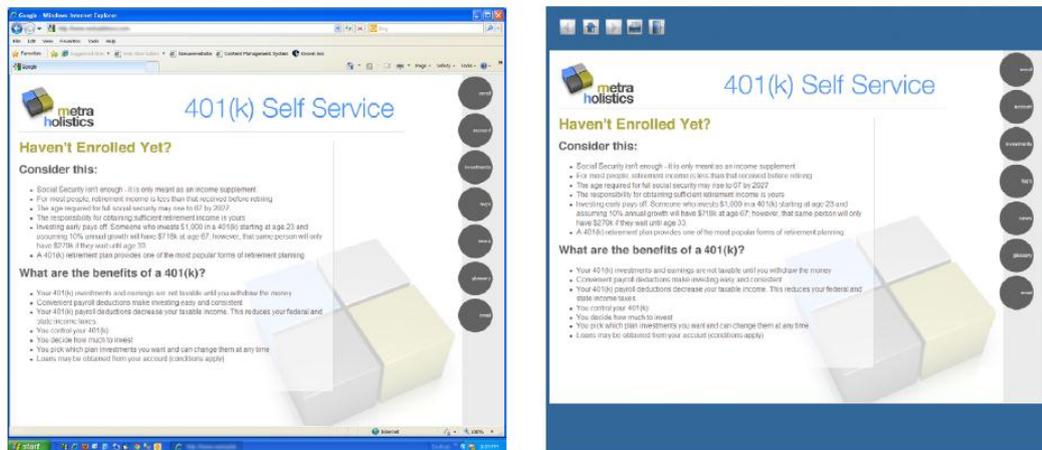


Figura 2-5: Comparación entre una aplicación vulnerable y una protegida.⁵

2.5.2 Deshabilitando las teclas especiales

Las teclas especiales y la combinación de estas pueden crear puertas para atacar al kiosco, debido a que permiten el acceso a ciertas funciones del sistema operativo. Por ejemplo, la combinación Ctrl-Alt-Supr permite el acceso al administrador de tareas y, mediante este, se puede apagar el sistema o ejecutar una aplicación. La combinación Alt-F4, por su parte, permite al usuario salir del programa, con lo que podría acceder a otras aplicaciones sin ningún control.

Todas estas acciones causadas por el uso de teclas especiales deben ser desactivadas para evitar acceso a aplicaciones, archivos e información. Se puede ir aún más allá, haciendo uso de las pantallas táctiles y teclados virtuales para no acceder a estas teclas, hasta el punto de descartar el uso de los teclados físicos. Estos teclados virtuales no poseerán teclas especiales o combinaciones de teclas, pues son propios de la aplicación. Un ejemplo de teclado virtual se muestra en la Figura 2-6.

⁵Gráfico tomado de (Slawsky)



Figura 2-6: Ejemplo de Teclado virtual⁶

2.5.3 Pantallas atractivas

Las pantallas atractivas son las interfaces gráficas que se muestran en el kiosco cuando no se encuentra en uso. Están diseñadas para llamar la atención del cliente hacia el kiosco. Cuando el usuario toca la pantalla, esta interfaz desaparece y se muestra el inicio de la aplicación. En este momento se debe iniciar un cronómetro que maneje un *time-out*. Transcurrido este tiempo, deberá aparecer una pregunta al usuario si necesita más tiempo para decidir. En caso de existir una acción por parte del cliente, regresará a su funcionamiento normal. Pero, en caso de que no haya interacción, significa que el cliente se ha ido y la aplicación debe ser reseteada.

2.5.4 Otras consideraciones de seguridad

Finalización de sesiones: Una vez que la aplicación ha completado su tarea, es importante que se cierre la sesión del usuario, se elimine toda la evidencia de la misma y se regrese a la pantalla principal, tal y como se indica en la Figura 2-4.

Control de cuadros de dialogo: Los cuadros de dialogo también representan oportunidades para atacar un kiosco. Un ejemplo de cuadro de dialogo es el de impresión, en el cual, un usuario podría pedir un número superior de copias, o seleccionar una menor cantidad de hojas, etc. Por esto, se deben manejar los

⁶ Gráfico tomado de (Slawsky)

cuadros de dialogo con cautela, eliminando aquellos cuyos parámetros son de riesgo.

Bloquear el acceso a ayudas: en una ayuda de Windows, el usuario puede abrir Internet Explorer y, con él, navegar a donde prefiera. Por lo que, un kiosco debe mantener desactivada esta opción.

2.6 Especificadores del Kiosco requerido para el proyecto

En este capítulo se describieron las aplicaciones que se puede dar a un kiosco interactivo, las partes que lo componen, los beneficios de su uso y, también, sus vulnerabilidades y como tratarlas. En base a las aplicaciones expuestas, los conceptos presentados, las partes enumeradas y los beneficios indicados se ha analizado que el kiosco requerido para el proyecto deberá tener las siguientes características:

- Pantalla táctil de 17" con un una resolución de 1280x1024.
- Puerto Ethernet y WiFi (para acceder a la red de la Universidad).
- Una impresora láser A4.
- CPU de 1.6GHZ o superior.
- Cabina.
- NO debe poseer teclado ni ratón.

En cuanto a las consideraciones de seguridad expuestas en este capítulo, deberán ser tomadas en cuenta al momento de diseñar la aplicación. En especial, junto con los lineamientos del capítulo siguiente, al momento de diseñar la interfaz gráfica del usuario. Para así, evitar estas vulnerabilidades y brindar mayor seguridad a la información de la Universidad del Azuay.

CAPÍTULO III

3. INTERFACES PARA PANTALLAS TÁCTILES (TOUCH SCREEN)

En este capítulo se describen los lineamientos para la creación de interfaces gráficas del usuario (GUI) para dispositivos con pantallas táctiles (touch screen). Estos lineamientos están basados en la tesis de maestría de (Sjöberg) y en la guía de diseño para aplicaciones táctiles de (Waloszek). Primero, se exponen algunas características a tomar en cuenta al diseñar interfaces táctiles. Posteriormente, se exponen las situaciones en la que se debe o no utilizar pantallas táctiles. Y, finalmente, se describen los lineamientos para la construcción de una interfaz gráfica de una pantalla táctil.

3.1 Características de las interfaces de las pantallas táctiles

Las pantallas táctiles son tradicionalmente utilizadas en público, en ambientes hostiles (p. ej. aeropuertos, estaciones de tren, museos, centros comerciales, etc.), en donde, los usuarios no tienen entrenamiento para utilizar la aplicación. Por ello, las pantallas táctiles son diseñadas para ser muy fáciles de usar y difíciles de confundirse, generando una interacción simple y rápida.

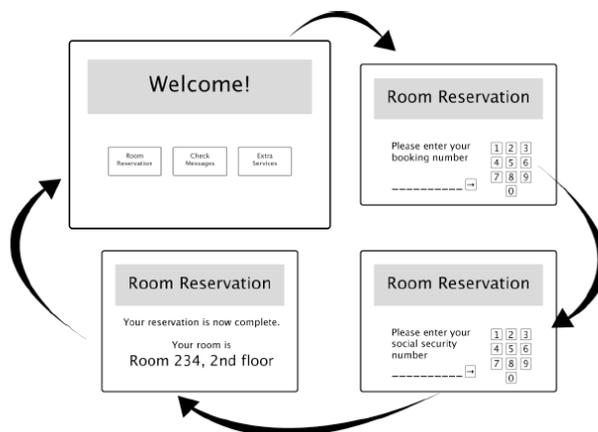


Figura 3-1: Ejemplo de Interfaz en forma de asistente⁷

⁷ Tomado de (Sjöberg)

En estos casos, la solución estándar es el uso de una interfaz semejante a un asistente, en donde, el usuario es guiado por medio de un proceso interactivo como se muestra en la Figura 3-1. Este proceso tiene mucha relación con el ciclo de interacción del software de la Figura 2-4, en especial, la parte de interacción entre el usuario y el kiosco.

Las pantallas táctiles pueden ser operadas con los dedos o por punteros tipo esfero. Gracias a ello, proveen una interacción directa (la más directa actualmente). Estas pantallas deben operar rápidamente en ciertas tareas y no deben requerir de entrenamiento al usuario, razón por la que sus aplicaciones deben ser diseñadas adecuadamente.

La eficiencia del uso de pantallas táctiles ha sido investigada en varias ocasiones y se ha probado que, en ciertas tareas, son más rápidas de usar que un mouse (interacción indirecta). Esto, sumado a la durabilidad de las pantallas táctiles, las hace adecuadas para estos ambientes demandantes. Sin embargo, la interacción directa de las pantallas viene con una limitación: la falta de retroalimentación táctil.⁸

La forma de interactuar con las pantallas táctiles introduce nuevos desafíos en el diseño, pues la interfaz gráfica del usuario (GUI Graphic User Interface) de una computadora de escritorio, no puede ser exportada directamente a una GUI de una pantalla táctil. Por lo cual, se debe tomar en consideración las siguientes características:

3.1.1 Tamaño del Objetivo

La mayor diferencia entre las GUIs de las pantallas táctiles y las comunes, o de controles indirectos (p. ej. Mouse), son las limitaciones físicas. Con un dispositivo indirecto se tiene mayor precisión al apuntar, mientras que el dedo humano, como

⁸ Traducido de la conclusión de (Sjöberg 17)

puntero, tiene “muy baja resolución” y es “difícil apuntar a objetos que son más pequeños que el ancho de un dedo.”⁹

Además del problema de la baja resolución de las yemas de los dedos, existe otro que hace aún más difícil apuntar objetos pequeños. Este problema consiste en la diferencia entre lo que el usuario desea tocar y lo que realmente toca. Esto se debe al paralelismo o espacio existente entre la pantalla táctil, que recepta los toques, y el display, que muestra en sí el contenido.

Como resultado de la combinación de los problemas anteriores, se ha concluido que los objetos deben tener un tamaño de 26mm para tener una precisión del 99.2%, y un tamaño de 20mm para una precisión del 95.6%.¹⁰ Sin embargo, el tamaño necesario puede ser menor si se usan monitores LCD o sensores táctiles incluidos en el display, ambos reducen el efecto causado por el paralelismo de las pantallas. Un ejemplo actual de esta reducción puede encontrarse en los dispositivos móviles totalmente táctiles, en donde, los botones son considerablemente pequeños en comparación del dedo humano.

3.1.2 Estrategia para apuntar en una pantalla táctil

El tamaño del objeto a seleccionar está relacionado con la estrategia para apuntarlo. Así como el mouse permite diferentes formas de interactuar con los objetos de la GUI (e. ej. Dar click, arrastrar, etc.), también existen diferentes técnicas para la interacción táctil.

Entre las distintas estrategias tenemos: *land-on* o toque, es la más simple y registra únicamente el toque inicial. La estrategia del primer contacto refina la *land-on*, permitiendo desplazar su dedo hasta el objetivo que desea seleccionar, es decir, se puede seleccionar un punto, distinto al inicial, con sólo desplazar el dedo hasta el

⁹ (Sjöberg 18)

¹⁰ (Sjöberg 19)

nuevo punto y ahí, levantar el dedo. Finalmente, existen estrategias que hacen uso de cursores para indicar qué objeto se está apuntando. Sin embargo, esta última estrategia debe ser descartada, pues, en muchas directrices sobre diseños de interfaces táctiles, se indica que se debe esconder el cursor, para que el usuario se enfoque en toda la pantalla y no sólo en la ubicación del cursor.

3.1.3 Teclados Virtuales

Muchos sistemas táctiles requieren de entradas de texto alternativas. Esto se debe a que, generalmente, no se utilizan teclados físicos al utilizar pantallas táctiles y, en el caso de los kioscos, se recomienda deshabilitar las teclas especiales. Una de las alternativas es utilizar el reconocimiento de caracteres, usado comúnmente en dispositivos con puntero como una PDA. Sin embargo, la alternativa más común y recomendada es el uso de teclados en pantalla o teclados virtuales.

Además de los teclados virtuales, se debe tomar en cuenta que la entrada de datos en las pantallas táctiles es secuencial, y que teclear números y letras con el dedo es agotador. Por ello, las pantallas táctiles deben ser utilizadas únicamente en lugares donde no se requiere mayor ingreso de texto o números.

En un estudio¹¹, se compararon diferentes tipos de teclados virtuales para determinar el de mayor eficiencia en la rapidez de ingreso de datos. Como resultado, se obtuvo que el teclado QWERTY tiene una mejor eficiencia en usuarios novatos, debido a que el teclado sí les es conocido.

Otro estudio¹², por su parte, compara el teclado virtual QWERTY con el teclado físico y la escritura por mouse en un teclado virtual. Como resultado se obtuvo 25.4, 58.2 y 17.1 palabras por minuto, respectivamente. El teclado físico permite el uso de ambas manos y, por tanto, permite mayor número de palabras por minuto

¹¹ (Mackenzie, Zhang y Soukoreff 235-244)

¹² (Sears 251-269)

que el teclado virtual (cuya entrada es secuencial y, por tanto, resulta cansado) y, por supuesto, mayor número que el mouse, debido a que es un dispositivo indirecto. Por esto, se reafirma que las pantallas táctiles deben ser utilizadas solamente en lugares donde no se requiere mayor ingreso de texto o números y, por tanto, se justifica el uso de un teclado virtual.

3.1.4 Ángulo de visión y ángulo de operación

Para minimizar los problemas de paralelismo (entre las pantallas: táctil y display), y disminuir la fatiga del brazo, es importante incorporar en el diseño de la aplicación táctil los ángulos de visión y de operación. Para ello, diferentes estudios han demostrado que el 92% de personas se ajustan a un ángulo entre 30 y 55 grados (Figura 3-2).

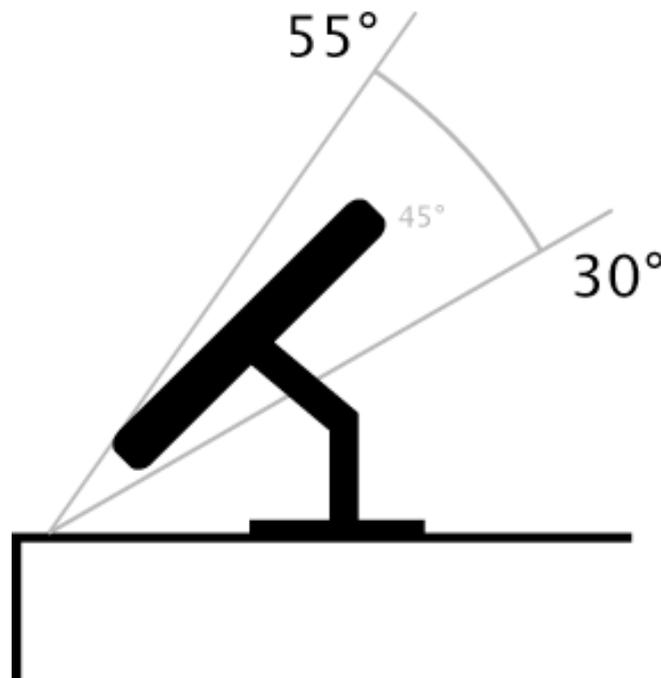


Figura 3-2: Ángulo entre la pantalla y la mesa¹³

¹³ Tomado de (Sjöberg)

3.1.5 Estrategia de desplazamiento

La necesidad de desplazamiento se origina por el tamaño limitado de las pantallas display. Esta limitante también se encuentra presente en los dispositivos táctiles, y, debido a que no existe mouse o teclado, el desplazamiento táctil es la única solución. Arrastrar de una manera táctil una barra de desplazamiento no es adecuado porque requiere precisión, y la mano puede obstruir la visión de la pantalla. Utilizar un botón para mover el contenido hacia arriba y abajo es una mejor solución, aunque ineficiente, porque requiere toques repetitivos.

En el caso de las pantallas táctiles, la mejor solución es el arrastrar el contenido en la dirección requerida, es decir, si el usuario desea mover el contenido hacia la derecha, lo desplaza con su mano hacia esa dirección. Esta estrategia se opone a la utilizada en las computadoras comunes, en donde se desplaza la barra hacia abajo para que el contenido suba. Aunque esta estrategia de mover el contenido directamente es atractiva, también puede producir que la mano obstruya la visión de la pantalla.

3.1.6 Movimiento de la mano y del brazo

Para que una GUI táctil sea eficiente, se debe evitar todo movimiento innecesario, pues conlleva dos cosas: incremento en el tiempo de respuesta y obstrucción de la pantalla. Para evitar estos movimientos innecesarios, los controles que se usan en conjunto deben ser agrupados.

Adicionalmente, se debe tomar en cuenta que el uso de las dos manos en la pantalla no necesariamente incrementa la eficiencia, pues esta depende de la naturaleza de la tarea a realizar. Sin embargo, un beneficio de esta técnica es la reducción del movimiento de las manos, pues cada una se puede enfocar en diferentes partes de la tarea.

3.1.7 Obstrucción de la visión de la pantalla

Como se expuso en las subsecciones anteriores, se debe tomar en cuenta que el uso de la mano como puntero causa que parte de la misma cubra la pantalla. Este inconveniente debe tomarse en cuenta en el diseño de GUI táctil, para que una parte pueda ser tapada y aún mostrar los datos importantes.

La única solución para la obstrucción de la pantalla es el diseño adecuado. Esto se logra mediante la reducción del movimiento de las manos, o minimizando las consecuencias de dicha obstrucción. Por ejemplo, las técnicas de desplazamiento y la posición de los botones afectan al movimiento de las manos. Al reducir el área que se debe recorrer para ir de una función a otra, se reduce la obstrucción de la pantalla.

3.1.8 Reacción o retroalimentación

Uno de los problemas de las pantallas táctiles es la falta de reacción o respuesta. Cuando se utiliza un teclado físico, existe una reacción física. El usuario puede sentir y escuchar cuando presiona un botón y, también, sentir donde se encuentran posicionados sus dedos. En cambio, en los teclados táctiles, para que el usuario conozca cual botón presionó, requiere de una respuesta por parte del software.

Las pantallas táctiles cuentan, como medios de respuesta, a los canales de sonido y, principalmente, los visuales. Para una interacción correcta con el usuario, el GUI táctil debe dejar en claro al usuario que acciones puede realizar y cuáles no. Y, en caso de realizar una acción, cuáles son sus consecuencias.

La obstrucción de la pantalla también debe ser considerada cuando se diseña la reacción, ya que, al presionar un botón pequeño, este puede quedar totalmente cubierto por la mano. Por ello, también se debe dar una respuesta adecuada (p. ej. mostrar la salida generada), en un lugar diferente al del botón presionado.

3.2 Usos de las pantallas táctiles

Las ventajas de las pantallas táctiles radican en que: permiten apuntar directamente a los objetos de la misma manera que en el mundo real, son rápidas y no necesitan teclado. Por ello, estas pantallas son recomendables en aplicaciones donde:

- No se puede realizar entrenamiento al usuario.
- La frecuencia de uso es baja.
- Se requiere poco o ningún ingreso de datos.
- Existe poco espacio.
- El ambiente puede ser “agresivo”.

Sin embargo, estas pantallas tienen sus desventajas como: la baja precisión de los dedos, la fatiga del brazo cuando se las utiliza por tiempos prolongados, la suciedad de la pantalla por las huellas de los dedos y la obstaculización de la visión por el uso de los brazos. Por estas razones, no es recomendable el uso de pantallas táctiles cuando:

- Se requiere entrenar a los usuarios.
- En ambientes de uso frecuente.
- Se requiere precisión.
- Se requiere ingresar grandes cantidades de datos.

3.3 Lineamientos para una interfaz táctil

Para la redacción de los lineamientos expuestos a continuación, se tomó en cuenta los lineamientos sugeridos por (Waloszek) y los enumerados por (Sjöberg). Los lineamientos de Sjöberg fueron seleccionados cuidadosamente, debido a que están dirigidos para crear GUIs táctiles de puntos de venta, es decir, para GUIs a ser

usadas por usuarios expertos. Sin embargo, muchos de estos lineamientos también pueden ser utilizados en la interfaz de un kiosco interactivo como el requerido para el sistema propuesto, y son los que han sido incluidos.

3.3.1 Reglas principales

- La aplicación debe correr rápido.
- El diseño debe ser lo más intuitivo posible.
- Se deben limitar el número de opciones.
- Guiar al usuario tanto como sea posible.
- Realizar pruebas de la aplicación mediante grupos focales, si en algún momento alguien realiza una pausa por una confusión, buscar la forma de cómo mejorar esa parte.

3.3.2 Usabilidad

3.3.2.1 Eventos

- Contacto (toque de la pantalla): produce problemas al querer cancelar la selección inicial.
- Al perder el contacto (levantar el dedo): permite cambiar la posición del dedo para cambiar o cancelar la selección inicial.
- No existen los eventos de *mouse-move* y *mouse-over*.

3.3.2.2 Interacción

- Clic: para iniciar acciones, seleccionar o especificar posiciones.
- Arrastrar: para definir caminos, arrastrar figuras, etc.
- Mantener seleccionado: para definir duraciones, ingresar texto, etc.

3.3.2.3 Operaciones básicas

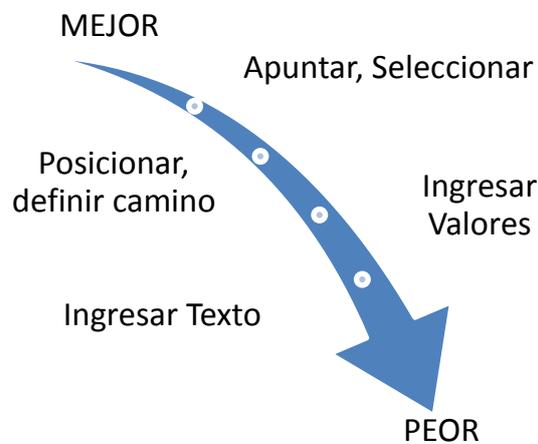


Figura 3-3: Operaciones básicas en un touch screen, organizadas de la mejor a la peor.

3.3.2.4 Cursores, punteros, pantallas de estado

- No utilizar cursores: los usuarios no pierden la posición de sus dedos.
- Utilizar una pantalla estática para demostrar que el sistema se encuentra activo.
- Proporcionar una reacción o respuesta en varias partes: No solamente confiar en el efecto 3D de los botones al ser presionados, ya que, estos efectos son obstaculizados por la mano del usuario. Además de estos efectos, se debe contar con una reacción que indique que información se le está proporcionando al sistema. (p. ej. Si se ingresan números, mostrar estos números en una parte que no esté bloqueada por la mano).

3.3.3 Diseño de la Pantalla

- La aplicación debe correr en pantalla completa y cubrir toda la pantalla.
- Utilizar *widgets* personalizados: Para liberar la mente de los usuarios, la interfaz táctil debe verse diferente a las tradicionales interfaces de escritorio. Los usuarios deben ver a la pantalla táctil como una “maquina interactiva”.

Con ello, debe resultar obvio que la GUI está diseñada para una interacción diferente a la común.

- Fondos: utilizar colores brillantes de fondo, que oculten las huellas dejadas en la pantalla. Los fondos estampados ayudan a que los ojos se enfoquen en la imagen y no en las reflexiones, incluso en áreas donde no existen íconos u opciones de menú.
- Diseñar la interfaz para el uso de cualquier mano: Para evitar la obstrucción de la pantalla y la fatiga del brazo, la interfaz debe permitir realizar operaciones con la mano izquierda o derecha. Sin embargo, no se debe diseñar una interfaz que sea eficiente únicamente con las dos manos.
- Utilizar una grilla para dividir la pantalla en un número determinado de celdas de un tamaño predeterminado. Después, identificar los grupos de elementos con funciones conjuntas, estos deben aparecer juntos y organizados por áreas funcionales. Estas áreas pueden ser indicadores de datos, elementos de ingreso o selección datos, botones de función, teclados virtuales, etc.
- Tener en cuenta el principio de la constancia en la ubicación: Al dividir la pantalla en áreas funcionales y colocar elementos en la pantalla, se debe mantener las áreas funcionales y los elementos de la interfaz en un mismo lugar. Como el espacio en la pantalla es limitado, se deberán intercambiar elementos dentro de una misma área funcional.

3.3.4 Gestión del espacio de la pantalla

- Utilizar pestañas: las pestañas permiten aumentar virtualmente el tamaño de la pantalla, dividiendo los contenidos en diferentes vistas que pueden ser accedidas mediante los botones de las pestañas.
- Utilizar áreas intercambiables: es una técnica similar al uso de pestañas, consiste en asignar ciertas áreas de la interfaz a funciones dadas como

visualización de datos, funciones, teclados, etc. Mediante pulsantes, se intercambian los elementos de cierta área. Por ejemplo, diferentes tipos de tablas, diferentes teclados, etc.

- Tomar en cuenta los cambios de pantallas: los usuarios pueden navegar entre pantallas, lo que les permite acceder a diferentes componentes, funciones o información. Sin embargo, esto puede causar confusión en los usuarios inexpertos. Por ello se debe:
 - Dar a las pantallas un diseño y aspecto consistente, para que los usuarios lo puedan percibir como una unidad o como un todo.
 - Dejar en claro qué opciones de navegación existen, p. ej. Como los usuarios pueden continuar o retroceder.
 - Limitar el cambio de pantallas a un mínimo.

3.3.5 Entrada de Datos

En las pantallas táctiles, la entrada de datos debe ser restringida a pequeñas cantidades. Por ello, existen diferentes técnicas para captar datos, como son: selección de valores, controles para incrementar o disminuir cantidades (Figura 3-4), o ingresando el valor directamente mediante un teclado virtual (generalmente agregados a pantalla bajo demanda).



Figura 3-4: Ingreso de datos mediante incremento y decremento de valores.¹⁴

¹⁴ (Waloszek 51)

La opción del teclado virtual (Figura 3-5) es la que requiere mayor esfuerzo por parte del usuario y no se recomienda su uso en sitios públicos. Sin embargo, es la técnica más usada y es la única opción para el ingreso del usuario y contraseña del estudiante, en el caso de la aplicación propuesta para emisión de certificados académicos.



Figura 3-5: Ejemplos de teclados virtuales: teclado numérico, teclado telefónico, y alfanumérico.¹⁵

Cuando es necesario el uso de teclados virtuales se debe:

- Utilizar un teclado virtual “económico”: El teclado debe ser escogido en base a la longitud esperada de los datos de entrada y, en base al espacio que el teclado ocupa.
- Reducir la cantidad necesaria de texto a introducir: El ingreso de datos en la pantalla táctil se lo realiza con un solo dedo, lo que hace que el ingreso sea lento y, por tanto, debe ser disminuido.
- Acomodar el teclado virtual para el ingreso de números: Convencionalmente se utiliza una disposición de 3x3 para el ingreso de datos numéricos. Dentro de esta distribución, existe la disposición de teléfono y la de teclado numérico de una computadora, como se observa a la izquierda de la Figura 3-5. El teclado numérico se utiliza con personas más experimentadas en el uso de computadoras.

¹⁵ (Waloszek 24)

- Reducir la cantidad de números a ingresar: En los teclados numéricos se utilizan varios dedos para introducir números a gran velocidad. Sin embargo, en una pantalla táctil, el ingreso es con uno solo, por ello, se debe evitar el ingreso de grandes cantidades de números.

3.3.6 Botones y menús

3.3.6.1 Botones

Las pantallas táctiles son principalmente utilizadas en aplicaciones basadas en la selección de objetos, la mayoría de ellas están llenas de botones y áreas para mostrar y, en ocasiones, ingresar información. Por ello, los botones juegan un papel importante en el diseño de una aplicación táctil y deben cumplir con las siguientes recomendaciones:

- Los botones deben tener un tamaño mínimo de 2x2cm y una separación de al menos 3mm entre ellos.
- Dividir la pantalla en una grilla que defina el tamaño básico de la celda para botones y otros elementos.
- Los botones deben presentar algún tipo de retroalimentación de su estado (activo/inactivo, prendido/apagado, etc.). Esta retroalimentación puede ser mediante efectos 3D al presionar el botón y/o cambios en otras partes de la pantalla.
- Utilizar tres diferentes indicadores de estado de los botones:
 - Icono predeterminado.
 - “Va a ser activado”: Representación que indique que el botón está siendo presionado (sólo cuando la acción se ejecute en el evento de soltar el botón).

- “Ha sido activado”: Señales de un pequeño periodo de tiempo que indiquen que el toque ha sido registrado y la correspondiente acción ha iniciado.
- Utilizar los botones para ofrecer opciones o alternativas, y para inicializar funciones.

3.3.6.2 Menús

Los menús consisten en un conjunto de elementos, de los cuales, el usuario puede seleccionar uno. Con frecuencia, los menús son representados mediante un grupo de botones. Estos botones pueden tener la apariencia de botones en sí o ser simplemente elementos gráficos (usualmente acompañados de etiquetas), como se muestra en la Figura 3-6.



Figura 3-6: Ejemplos de menús con botones. A la izquierda con apariencia de botones y diferenciando los botones y menús mediante formas y estilos. A la derecha con imágenes y ocultando la apariencia de botones.¹⁶

Para la creación de menús con botones se debe seguir las siguientes recomendaciones:

- Los botones y los menús (representados con botones) deben estar claramente separados y diferenciados para que no existan errores al realizar la selección.

¹⁶ (Waloszek 28)

- Los botones utilizados también serán manipulados por los dedos, por ello, deben cumplir con los mismos lineamientos descritos anteriormente.
- Agrupar los botones del menú mediante los métodos para agrupar elementos en la pantalla (p. ej. Usando frames, formas, colores, colores de fondo, texturas etc.), para que los botones sean vistos como parte de un todo.

Para la creación de menús, también se pueden utilizar listas, ordenadas de arriba abajo, o de izquierda a derecha en caso de que sean pocos datos. Sin embargo, este tipo de menús son más difíciles de explorar visualmente, por ello, deben ser usados solamente cuando hay poco espacio en pantalla y no exista otro tipo de presentación mejor.

3.3.7 Desplazamiento y Navegación

En las pantallas táctiles es más fácil realizar un desplazamiento aplastando botones que desplazando una barra. Por esta razón, no se recomienda el uso de barras de desplazamiento, incluso si son de mayor tamaño. Lo que se recomienda es el uso de botones de desplazamiento (Figura 3-7).

Los botones de desplazamiento suelen ser utilizados de forma repetitiva, por lo que se recomienda que se repita la acción de desplazamiento después de un tiempo, mientras el botón siga presionado.

Los botones de desplazamiento no deben ser utilizados para desplazarse en la pantalla en sí, sino para desplazarse o navegar dentro de un contenido mostrado en una pantalla táctil (p. ej. Para cambiar de página un documento). Lo óptimo en un diseño es no requerir de ningún tipo de desplazamiento.

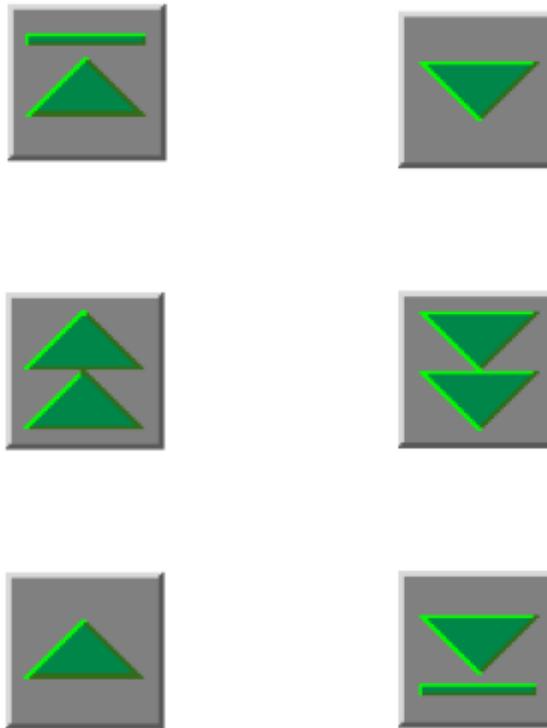


Figura 3-7: Botones de desplazamiento hacia arriba (izquierda) y hacia abajo (derecha).¹⁷

3.3.8 Recomendaciones Generales

- Diseñar los objetos con un tamaño razonable: Dependiendo del tipo de objeto, el tamaño puede variar, pero aquellos que son más pequeños que las yemas requieren mayor precisión.
- Usar una estrategia intuitiva para apuntar objetos: De preferencia utilizar la estrategia de respuesta al levantar el dedo, pues permite un pequeño ajuste en caso de fallo. No se debe utilizar desplazamiento, doble clic, menús desplegables, ventanas múltiples u otros elementos que puedan confundir al usuario común e interferir con la usabilidad y eficiencia.
- Utilizar un tipo de evento en cada objeto: No mezclar eventos en un mismo objeto. Si un botón es táctil y arrastrarle, la baja precisión del dedo puede provocar la invocación de la acción incorrecta.

¹⁷ (Waloszek 43,44)

- Cumplir las expectativas del usuario cuando se desplace el contenido: El desplazamiento debe ser intuitivo y eficiente. El desplazamiento debe ser accionado de una manera continua, similar a arrastrar un objeto en la vida real. Además, se debe asegurar que el usuario pueda realizar el desplazamiento sin obstruir su visión del contenido.
- Mostrar la información donde se encuentra la atención del usuario: Exponer la información, especialmente la crítica, donde está dirigida la atención del usuario.
- Crear una interfaz que solicite confirmación: solicitar la confirmación de una selección realizada cuando las consecuencias sean destructivas o difíciles de deshacer.
- Utilizar metáforas: las metáforas proveen a los usuarios un ambiente familiar en la pantalla (p. ej. Utilizar una imagen con una puerta y una flecha para indicar que se desea abandonar el programa). Esto permite que los usuarios aprendan el uso de la aplicación de una manera más rápida.

En este capítulo se presentaron las características que diferencian a las pantallas táctiles de las comunes, principalmente en diseño de la interfaz gráfica y la usabilidad. Tomando en cuenta estas características, se especificaron los casos en que es recomendable el uso de pantallas táctiles y en cuales se debe optar por otra opción. Así también, en base a estas características, se presentaron los lineamientos que se deben tomar en cuenta al momento de diseñar interfaces graficas de usuario para estas pantallas. Se hizo especial énfasis en el diseño de los botones y menús, del desplazamiento del contenido, del espacio de la pantalla y en la usabilidad. Estos lineamientos son los utilizados posteriormente en diseño de la GUI del sistema propuesto, que se presenta en el Capítulo VI.

CAPÍTULO IV

4. WEB SERVICES

El presente capítulo presenta el marco teórico referente a Web Services y la arquitectura orientada a servicios, sus conceptos, tecnologías y consideraciones de seguridad. El capítulo está basado principalmente en el documento de la W3C (Web Services Architecture), pero también incorpora otras fuentes que permiten complementar la información.

4.1 Definición

Un web service o servicio web es un software diseñado para permitir la comunicación entre máquinas sobre una red. Sus interfaces y enlaces pueden ser definidos, descritos y descubiertos como artefactos XML. Habitualmente, la interfaz de conexión está descrita en un formato XML llamado WSDL. Un web service soporta interacciones directas con otros agentes software mediante el intercambio de mensajes descritos en XML (mensajes SOAP), y transmitidos por protocolos basados en internet. Generalmente, los mensajes son transmitidos por HTTP en conjunto con otras normas WEB.

En otras definiciones existentes se incluye los estándares XML (etiquetado), SOAP (transmisión), WSDL (descripción) y UDDI (servicios disponibles) como parte de la definición de un web service. Sin embargo, estos estándares, aunque son los más usados, no constituyen la esencia de la tecnología. De hecho, los problemas subyacentes a web services son los mismos sin importar los estándares usados.¹⁸

¹⁸ (Alonso, Casati y Kuno 124)

4.1.1 Agentes y Servicios

Un agente es una porción de software o hardware que envía y recibe mensajes. Así, un web service debe ser implementado por un agente en concreto. En cambio, un servicio es el recurso que presta las distintas funcionalidades. Por ejemplo, al implementar un web service se puede optar por usar diferentes lenguajes de programación (agentes), para brindar una misma funcionalidad (servicio).

4.1.2 Solicitantes y proveedores

Los solicitantes y proveedores pueden ser entidades (personas u organizaciones) o agentes. La entidad proveedora es la que provee el agente para implementar un servicio. La entidad solicitante es la que espera hacer uso del web service de la entidad proveedora. Esta hará uso de su agente solicitante para intercambiar mensajes con el agente proveedor.

Generalmente, el agente solicitante es el que inicia el intercambio de mensajes, aunque no es siempre cierto. Sin embargo, se utiliza esta nomenclatura aunque el agente proveedor sea el que inicie el intercambio.

4.1.3 Descripción del Servicio

Los mecanismos para intercambio de mensajes están documentados en una descripción del servicio web, WSD (Web Service Description). La WSD es una especificación de la interfaz del web service, escrita en el lenguaje WSDL (Web Service Description Language). La WSD define el formato de los mensajes, tipos de datos, protocolos de transporte y formatos de transporte en serie, que deben ser usados entre los agentes solicitante y proveedor. Además, especifica los puntos de red desde donde se puede invocar al agente proveedor e información sobre los modelos esperados para el intercambio de mensajes.

4.1.4 Semántica

Mientras la descripción del servicio representa un contrato sobre los mecanismos de interacción con un servicio, la semántica representa un contrato que rige el sentido y propósito de la interacción. Mientras se utilicen lenguajes más ricos en semántica, más información pasará de la semántica a la descripción del servicio.

4.1.5 Conexión con un web service

Para conectarse con un web service, genéricamente, se necesitan los siguientes pasos (Figura 4-1):

- 1) La entidad solicitante y proveedora empiezan a “conocerse”.
- 2) Las entidades aceptan la descripción del servicio y la semántica que gobernarán la comunicación entre sus agentes. (Normalmente la solicitante aceptará lo ofertado por la proveedora).
- 3) El WSD y semántica son captadas por los agentes.
- 4) Los agentes intercambian mensajes, ejecutando una tarea solicitada por las entidades.

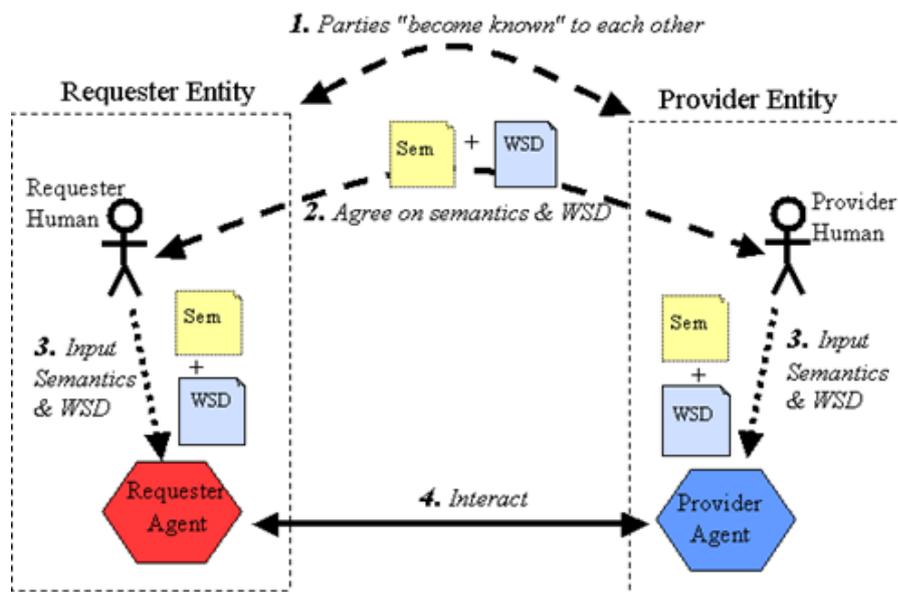


Figura 4-1: El proceso de conexión con un web service.¹⁹

¹⁹ (Web Services Architecture 9)

4.2 Necesidad de web services

Los web services fueron desarrollados para cumplir un objetivo en especial, el de ser puntos de acceso a la información del sistema. De esta manera, el uso dado a los web services es el de “exponer la funcionalidad de los sistemas internos y hacerlos accesibles mediante la web, pero de una manera controlada.”²⁰ Los web services son como envoltorios o *wrappers* que “encapsulan una o varias aplicaciones, proveyendo una interfaz única y un servicio web.” Se debe tomar en cuenta que encapsular los componentes y esconder su heterogeneidad (poseer componentes homogéneos), es la “clave para la integración de las aplicaciones.” Por lo tanto, los web services *wrappers* y homogéneos.

Los servicios web se utilizan para permitir el acceso a la información de una manera más general. Por esta razón, son una ayuda a la tecnología para la integración de aplicaciones empresariales (EAI Enterprise application integration), la cual tiene como objetivos: integrar los datos y procesos, manteniendo una separación entre las reglas del negocio y las aplicaciones; y proporcionar una interfaz única de acceso para un conjunto aplicaciones. Los Middleware convencionales, a diferencia de los web services, requieren que los consumidores del servicio se adapten a un middleware específico del proveedor del servicio. Por tanto, los servicios web resultan una opción más atractiva.

Aunque la mayoría de la literatura se enfoca en el uso de servicios web para aplicaciones e integración B2B (business-to-business), los web services no necesitan ser accedidos por Internet, es posible “poner a disposición los servicios web a los clientes dentro de una red LAN”²¹, es decir, para facilitar una integración entre las aplicaciones internas de la compañía, y no sólo intercambios entre compañías. Cuando se utiliza los web services para servicios dentro de una

²⁰ (Alonso, Casati y Kuno 134)

²¹ (Alonso, Casati y Kuno 135)

compañía, estos, en muchos casos, funcionan como un middleware de un nivel más alto, que encapsula los servicios proporcionados por otros middleware para que sean accedidos mediante el servicio web.

El caso que se presenta en la presente tesis, corresponde a poner a disposición los servicios web dentro de la Universidad, para que sirvan como un sistema de integración entre sus aplicaciones. En este documento, se desarrolla un Sistema para la emisión de Certificados Académicos, haciendo uso de los servicios desarrollados. Sin embargo, estos quedarán disponibles para su futuro uso en: consulta de calificaciones mediante dispositivos móviles, solicitudes y servicios para los estudiantes mediante el portal web, etc.

4.3 Arquitectura Orientada a Servicios (SOA)

SOA (Service Oriented Architecture) es una arquitectura de software que define la utilización de servicios, para dar soporte a los requerimientos del software del usuario. Los nodos de la red ponen sus recursos a disposición de otros como servicios independientes, a los que tienen acceso de un modo estándar. Normalmente se utilizan servicios web (con SOAP y WSDL), pero se puede implementar SOA con cualquier tecnología orientada a servicios.²²

A diferencia de la orientación a objetos, las SOAs están formadas por servicios poco acoplados y altamente interoperables.

4.3.1 Sistemas Distribuidos

Un sistema distribuido consiste en un conjunto de agentes de software discretos y diversos que deben trabajar en conjunto para llevar a cabo una tarea. Estos agentes no operan dentro del mismo ambiente, se comunican mediante protocolos de red a nivel de hardware o software. Lo que significa que la comunicación es de

²² (Cortizo)

menor velocidad y confianza que en una arquitectura tradicional. Por ello, las arquitecturas distribuidas requirieren consideraciones sobre la latencia en acceso remoto, los problemas de concurrencia y las posibilidades de fallo.

SOA es una arquitectura de sistemas distribuidos que típicamente se caracteriza por las siguientes propiedades:

- Vista lógica: El servicio es una vista abstracta y lógica de: los programas, bases de datos y reglas del negocio existentes. Está definido desde el punto de vista de “que es lo que hace”.
- Orientado a mensajes: El servicio está definido en términos de intercambio de mensajes entre los agentes solicitante y proveedor. La estructura interna de un agente es abstraída de la arquitectura SOA, es decir, no es necesario conocer como un agente implementa el servicio prestado.
- Orientado a la descripción: El servicio se encuentra descrito en una metadata procesable por computadora. En donde, solamente los datos que pueden ser expuestos al público y son importantes se incluirán en la descripción. La semántica del servicio puede estar o no documentada en la descripción.
- Granularidad: Los servicios generalmente realizan pocas operaciones, con mensajes relativamente grandes y complejos.
- Orientado a las redes: Los servicios están orientados a ser utilizados en una red, aunque no es obligatorio.
- Independiente de la plataforma: Los mensajes son enviados a través de interfaces, en un formato estandarizado e independiente de la plataforma, para facilitar la interoperabilidad. Por esta razón se utiliza XML.

4.3.2 Servicios Web y estilos Arquitectónicos

Los sistemas distribuidos tienen algunos problemas que deben considerarse en la solución:

- Problemas causados por la latencia y confiabilidad el transporte.
- Ausencia de memoria entre el solicitante y el objeto.
- Problemas generados por casos de fallo.
- Los accesos concurrentes.

Es importante mencionar que estos problemas se deben tomar en cuenta en todos los sistemas distribuidos, ya sean estos implementados usando COM/CORBA o tecnologías Web services. Sin embargo, el uso de web services no siempre es la mejor alternativa para implementar SOAs. Si existe experiencia en la implementación de COM/CORBA y no se necesita una plataforma neutral, los beneficios de SOAP/WSDL no justifican su uso frente a sus costos en rendimiento.

Por ello, el uso de SOA y web services son más apropiados en los siguientes casos:

- El servicio debe operar sobre Internet, donde la confiabilidad y velocidad no están garantizadas.
- Cuando no hay capacidad para administrar la implementación en la que todos los proveedores y solicitantes se actualicen en conjunto.
- Cuando los componentes de sistemas distribuidos corren en diferentes plataformas y poseen productos de diferentes proveedores.
- Cuando una aplicación necesita ser usada sobre una red y puede ser empaquetada como un servicio web.

El sistema propuesto, se asocia con el último caso expuesto, pues el Kiosco se conectará al servidor de una manera remota mediante red local. Esto resultará especialmente útil para las facultades que se encuentran fuera del campus principal. Otra de las razones para optar por el uso de SOA con web services, es su portabilidad. Los servicios web constituirán un middleware para obtener los datos para la implementación de futuros sistemas, como consulta de calificaciones mediante Smartphones e inclusive futuras exportaciones del sistema de certificados a la web con el uso de firmas digitales.

4.3.3 Arquitecturas REST

Roy Fielding propuso un estilo de arquitectura para aplicaciones web confiables conocidas como *Representation State Transfer (REST)*. REST web es un subconjunto de WWW, en el cual, los agentes proveen una semántica de interfaz uniforme (*GET, PUT, POST y DELETE*), en vez de interfaces arbitrarias o específicas de la aplicación. Las interacciones REST son *stateless* (estado intrínseco), en el sentido de que el significado de un mensaje no depende del estado de la conversación.

Los web services REST no utilizan los estándares SOAP, ni WSDL. En su lugar, el intercambio de mensajes puede realizarse en cualquier formato (XML, JSON, HTML, etc). Estos web services son útiles para aplicaciones que no requieren seguridad aparte de la disponible en la infraestructura HTTP, y en donde esta puede ser usada.²³

De manera general existen 2 grandes tipos de web services:

- **REST-compliant web services**, en el cual, el propósito es manipular representaciones XML de los recursos usando operaciones *stateless*, es decir, obedecen la arquitectura REST.

²³ (Introduction to Web Services)

- **Web services arbitrarios**, en los cuales el servicio puede exponer un conjunto arbitrario de operaciones.

Ambos tipos utilizan URIs (Identificador uniforme de recurso), para identificar los recursos, y protocolos web (tales como HTTP y SOAP) y XML para el intercambio de mensajes.

4.4 Tecnologías Web Services

La arquitectura de los web services utilizan muchas tecnologías interrelacionadas y superpuestas. Hay muchas maneras de visualizar estas tecnologías, así como hay muchas maneras de construir y usar Web Services. La Figura 4-2 muestra algunas familias de estas tecnologías.

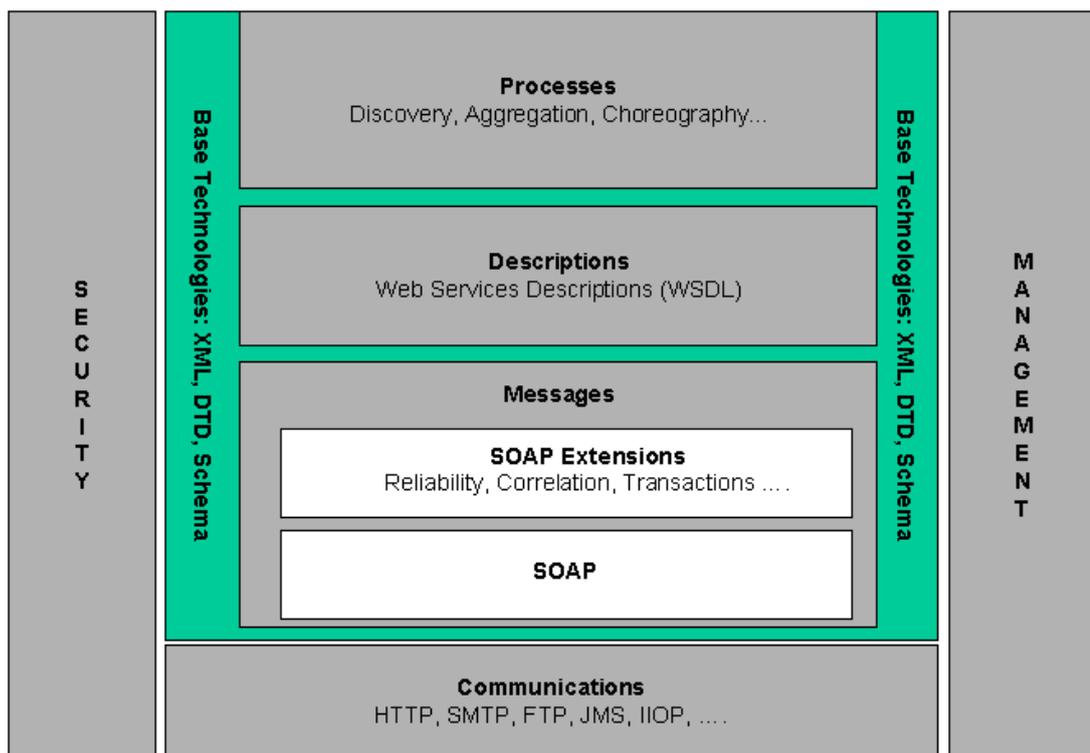


Figura 4-2: Pila de la Arquitectura Web Service²⁴

A continuación se describen algunas tecnologías que tienen estrecha relación con esta tecnología.

²⁴ (Web Services Architecture 64)

4.4.1 XML

XML es una tecnología clave para el uso de web services y es utilizada en varias partes de la arquitectura. XML ofrece un estándar flexible, genérico y extensible de formato de datos, el cual, reduce la carga de implementación de muchas tecnologías para asegurar el éxito de los Web services.

Para propósitos de la arquitectura, los conceptos importantes de XML son el infoset, el esquema y los namespaces.

XML infoset: describe el modelo abstracto de datos de un documento XML, en términos de un conjunto de elementos de información. Su propósito es proveer un conjunto de definiciones que serán usadas en otras especificaciones, que necesitan hacer referencia a la información de un documento XML formado correctamente.²⁵

XML Schema (XSD): es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. La Figura 4-3 indica un ejemplo de un esquema para una dirección con sus respectivos tipos de datos.²⁶

XML Namespaces: se utilizan para proveer un nombre único a elementos y atributos en un documento XML, ya que, una instancia XML puede contener nombres de elementos o atributos de más de un vocabulario XML. Si a cada vocabulario se le da un namespace, se resuelve posibles ambigüedades por nombres idénticos. Para ello, se utiliza el atributo xmlns para declarar el nombre del namespace. Ej, xmlns:html='http://www.w3.org/1999/xhtml'. Al momento de utilizarlo se lo indica el *nombre_namespace:elemento*. Ej. <html:head>.

²⁵ (XML-infoset)

²⁶ (W3C, Extensive Markup Language (XML))

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Direccion">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Persona" type="xs:string" />
        <xs:element name="Casa" type="xs:string" />
        <xs:element name="Calle" type="xs:string" />
        <xs:element name="Calle" type="xs:string" />
        <xs:element name="País">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="IN" />
              <xs:enumeration value="DE" />
              <xs:enumeration value="EC" />
              <xs:enumeration value="UK" />
              <xs:enumeration value="US" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Figura 4-3: Ejemplo XML Schema

4.4.2 SOAP

SOAP 1.2 provee un framework estándar y extensible para empaquetar e intercambiar mensajes XML. En el contexto de la arquitectura, SOAP 1.2 provee, además, un mecanismo conveniente para referirse a los servicios prestados (mediante cabeceras).

En SOAP 1.2 Parte 1²⁷, se define un framework de mensajes basado en XML. Los mensajes SOAP pueden ser transmitidos por una variedad de protocolos de red; como HTTP, SMTP, FTP, o cualquier protocolo propietario de mensajería.

²⁷ (SOAP Version 1.2 Part 1: Messaging Framework (Second Edition))

En SOAP 1.2 Parte 2²⁸, se definen 3 componentes opcionales: un conjunto de reglas de decodificación para expresar instancias de tipos de datos definidos por la aplicación, un acuerdo para representar llamadas remotas y respuestas, y un conjunto de reglas para usar SOAP con HTTP 1.1.

La definición que nos interesa de SOAP es la ligada al protocolo SOA. Esta definición expresa que un mensaje SOAP representa la información necesaria para invocar un servicio (Figura 4-4), o reflejar los resultados de la invocación de un servicio, conteniendo la información especificada en la interfaz de definición del servicio (Figura 4-5).

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.example.com/ws">
      <productID>827635</productID>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

Figura 4-4: Ejemplo de invocación de servicio mediante SOAP.²⁹

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productName>Toptimate 3-Piece Set</productName>
        <productID>827635</productID>
        <description>3-Piece luggage set. Black Polyester.</description>
        <price currency="NIS">96.50</price>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

Figura 4-5: Ejemplo de respuesta de servicio mediante SOAP.¹³

²⁸ (SOAP Version 1.2 Part 2: Adjuncts)

²⁹ (Cortizo)

4.4.3 WSDL

Web services description lenguaje (WSDL)³⁰ provee un modelo y formato XML para la descripción de servicios web. WSDL describe un servicio web en dos etapas: una abstracta y una concreta.

En el nivel abstracto, se describe el servicio web en términos de los mensajes que se envían y reciben. Los mensajes son descritos de una manera independiente de un formato de conexión específica. Generalmente mediante XMLSchema.

En el nivel concreto, un enlace especifica los detalles de los formatos de transporte para una o más interfaces. Un endpoint asocia una dirección de red con un enlace. Y, finalmente, un servicio agrupa los endpoints que implementan una interfaz común.

Las definiciones pueden ser mapeadas a cualquier lenguaje de implementación, plataforma, modelo de objetos o sistema de mensajes. Mientras el remitente y receptor estén de acuerdo en la descripción del mensaje (WSDL), la implementación puede ser cualquiera.

Un documento WSDL 1.1 usa los siguientes elementos en la definición de servicios en la red:³¹

- *Types*: es un contenedor de definiciones de tipo de datos, es decir, describe los datos usando algún tipo de sistema. (p.ej. XSD)
- *Message*: es una descripción abstracta de los datos que son transferidos. Normalmente corresponde a una operación. Además, Message contiene la información necesaria para realizar la operación. Cada mensaje contienen una o más partes lógicas (parts). Cada una está asociada con un elemento

³⁰ (Web Services Description Language (WSDL))

³¹ (Web Services Description Language (WSDL))

(element) y un tipo de dato (type) que pueden estar definidos dentro del XSD.

- *Operation*: es una descripción abstracta de una acción soportada por un servicio. Define las acciones SOAP y la forma en la que el mensaje es codificado. Una operación es como un método o función llamado en un lenguaje de programación normal.
- *Port type*: es un conjunto de operaciones soportadas por uno o más *endpoints*. Define al servicio web, las operaciones que puede ejecutar, y los mensajes que serán usados para realizar cada operación.
- *Binding*: define el formato de los mensajes y los detalles del protocolo, para operaciones y mensajes definidos en un *port type* específico.
- *Port*: Define la dirección o punto de conexión a un único *endpoint*. Típicamente está representado por una dirección URL para cada *binding*.
- *Service*: una colección de *endpoints (ports)* relacionados, es decir, las funciones que han sido expuestas mediante protocolos web.

La Figura 4-6 presenta un ejemplo de WSDL que ejemplifica cada uno de estos términos para mejorar su comprensión.

```

<?xml version="1.0"?>
<!-- root element wsdl:definitions defines set of related services -->
<definitions name="StockQuote"
  targetNamespace="http://example.com/stockquote.wsdl"
  xmlns:tns="http://example.com/stockquote.wsdl"
  xmlns:xsd="http://example.com/stockquote.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  <!-- wsdl:types encapsulates schema definitions of communication
types; here using xsd -->
  <wsdl:types>

    <!-- all type declarations are in a chunk of xsd -->
    <xsd:schema targetNamespace="http://example.com/stockquote.xsd"
      xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">

      <!-- xsd definition: TradePriceRequest [... tickerSymbol
string ...] -->
      <xsd:element name="TradePriceRequest">
        <xsd:complexType>
          <xsd:all>
            <xsd:element name="tickerSymbol" type="string"/>
          </xsd:all>
        </xsd:complexType>
      </xsd:element>

      <!-- xsd definition: TradePrice [... price float ...] -->
      <xsd:element name="TradePrice">
        <xsd:complexType>
          <xsd:all>
            <xsd:element name="price" type="float"/>
          </xsd:all>
        </xsd:complexType>
      </xsd:element>

    </xsd:schema>
  </wsdl:types>

  <!-- request GetLastTradePriceInput is of type TradePriceRequest -->
  <wsdl:message name="GetLastTradePriceInput">
    <wsdl:part name="body" element="xsd1:TradePriceRequest"/>
  </wsdl:message>

  <!-- request GetLastTradePriceOutput is of type TradePrice -->
  <wsdl:message name="GetLastTradePriceOutput">
    <wsdl:part name="body" element="xsd1:TradePrice"/>
  </wsdl:message>

  <!-- wsdl:portType describes messages in an operation -->
  <wsdl:portType name="StockQuotePortType">

    <!-- the value of wsdl:operation eludes me -->
    <wsdl:operation name="GetLastTradePrice">
      <wsdl:input message="tns:GetLastTradePriceInput"/>
      <wsdl:output message="tns:GetLastTradePriceOutput"/>
    </wsdl:operation>
  </wsdl:portType>

```

```

<!-- wsdl:binding states a serialization protocol for this service -->
  <wsdl:binding name="StockQuoteSoapBinding"
    type="tns:StockQuotePortType">

    <!-- leverage off soap:binding document style (no wsdl:foo
    pointing at the soap binding) -->
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>

    <!-- semi-opaque container of network transport details classed
    by soap:binding above -->
    <wsdl:operation name="GetLastTradePrice">

      <!-- again bind to SOAP? -->
      <soap:operation
        soapAction="http://example.com/GetLastTradePrice"/>
      <!-- further specify that the messages in the wsdl:operation
      "" use SOAP? -->
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

  <!-- wsdl:service names a new service "StockQuoteService" -->
  <wsdl:service name="StockQuoteService">
    <wsdl:documentation>My first service</documentation>

    <!-- connect it to the binding "StockQuoteBinding" above -->
    <wsdl:port name="StockQuotePort"
      binding="tns:StockQuoteBinding">

      <!-- give the binding an network address -->
      <soap:address location="http://example.com/stockquote"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

Figura 4-6: Ejemplo de WSDL para la operación GetLastTradePrice del servicio StockQuoteService.

La versión WSDL 2.0 tiene unos pocos cambios con respecto a WSDL 1.1, entre los que se encuentran:

- Más semántica en la descripción del lenguaje.
- Se eliminaron las construcciones de mensajes.
- La sobrecarga de operadores no es compatible.
- PortTypes se renombran a interfaces.
- Ports se renombran a endpoints.

La Figura 4-7 indica la estructura de WSDL 1.1 y sus diferencias con 2.0

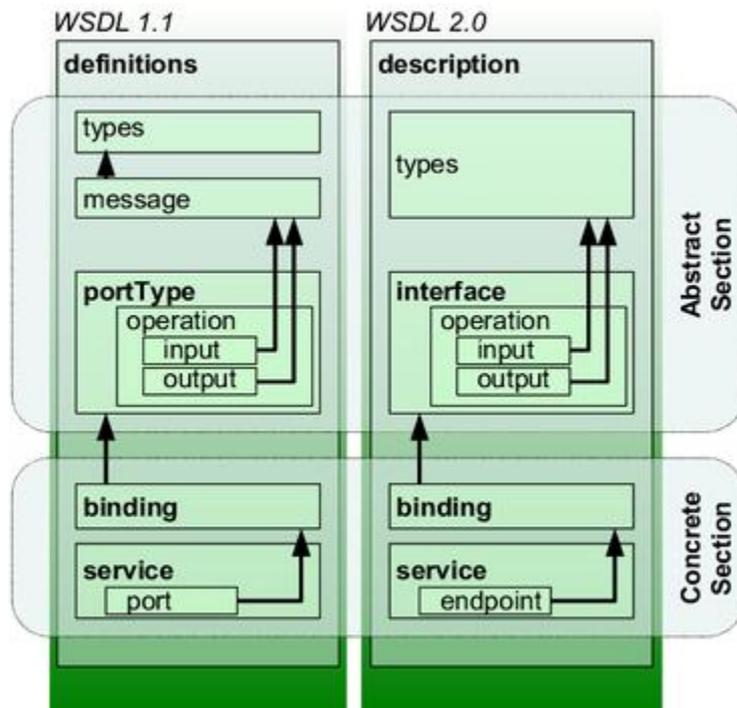


Figura 4-7: Comparación y analogías entre WSDL 1.1 y 2.0³²

Sin embargo, para el caso del sistema emisor de certificados académicos, se utilizará la especificación WSDL 1.1, debido a que es la utilizada por el framework a utilizar JAX-WS.

4.5 Utilizando Web Services

La Figura 4-1 ilustra los pasos para enganchar un web service. Estos pasos son necesarios, pero no siempre suficientes. En muchos casos, serán necesarios pasos adicionales y, en otros, el orden de los mismos puede variar. A continuación se presenta una especificación más detallada de estos pasos:

1. Las entidades se “conocen”, en este sentido la parte que inicia la interacción toma conciencia de la existencia de la otra parte. Para esto hay dos casos:
 - a. El agente solicitante es el que inicia la acción y debe tomar conciencia de la existencia de la entidad proveedora. Esto puede ocurrir de dos

³² (WSDL)

maneras: Obteniendo la dirección del agente proveedor directamente de la entidad proveedora. O, utilizando un servicio que permita localizar la descripción del servicio, de una manera manual o autónoma.

- b. El agente proveedor inicia el intercambio de mensajes. El que las entidades se “conozcan” significa que la entidad proveedora toma conciencia de la entidad solicitante, es decir, de alguna manera obtiene la dirección del agente solicitante. El como ocurre depende de la aplicación y no tiene importancia en la arquitectura.
2. La entidad que solicita y la que provee deben estar de acuerdo en la descripción del servicio y la semántica que gobernará su interacción. El estar de acuerdo significa que ambas partes deben tener entendimientos similares de la descripción del servicio y su semántica. Hay distintas maneras de lograrlo:
- a. Las entidades se deben comunicar directamente para, explícitamente, estar de acuerdo con la descripción del servicio y la semántica.
 - b. La entidad proveedora puede publicar y ofertar la descripción y semántica como contratos de “tómalo o déjalo”, en donde el solicitante debe aceptar, sin cambios, estas condiciones.
 - c. La descripción y semántica pueden ser definidos como un estándar por una organización, y puede ser utilizada por muchas entidades solicitantes y proveedoras. En este caso, la entidad solicitante y la proveedora deben aceptarlas de una manera individual.
 - d. El solicitante puede publicar la descripción del servicio y la semántica; y ofertarla a las entidades proveedoras como un “tómalo o déjalo”. Esto sucede cuando una compañía requiere de otras para que le provean servicios web que se acoplen a la descripción y semántica existentes.

3. La descripción del servicio y la semántica son una entrada para los agentes, es decir, la información anterior debe ser usada o implementada por los agentes. Lo cual, se puede lograr de varias formas:
 - a. Un agente puede ser forzado a implementar una descripción y semántica específicas.
 - b. Un agente puede ser codificado de una manera más general para aceptar dinámicamente descripciones y semántica.
 - c. Un agente puede ser creado en primera instancia y luego generar la descripción del servicio y su semántica a partir de él.

Sin importar la implementación, los dos agentes deben usar o implementar la descripción y su semántica antes de interactuar.
4. Los agentes intercambian mensajes SOAP, ejecutando la tarea solicitada por las entidades.

4.6 Seguridad en Web services

Una amenaza en un web service conlleva a una amenaza al sistema y toda su estructura. “La debilidad principal de los Web Services está dada por el modelo de interacción entre el consumidor y el proveedor del servicio.”³³ Esto se debe a que el proveedor describe su servicio utilizando WSDL, dando así, una información completa de los parámetros que espera recibir y, por tanto, una forma de atacar.

Para asegurar los web services se necesitan algunos mecanismos de seguridad basados en XML, que resolverán problemas relacionados con la autenticación, control de acceso, ejecución de políticas de seguridad distribuidas, seguridad a nivel de mensajes, etc.

³³ (Seguel)

Las implementaciones de web services pueden requerir de mecanismos de seguridad punto a punto (point-to-point) y/o consumidor a proveedor (end-to-end), dependiendo de la amenaza. Los mecanismos de seguridad tradicionales, orientados a conexión y punto a punto, pueden no bastar para cumplir con los requisitos de seguridad del mecanismo end-to-end. Sin embargo, las consideraciones de seguridad deben ser en base a un balance costo/beneficio.

4.6.1 Políticas de seguridad

Desde el punto de vista de la arquitectura existen 3 conceptos fundamentales en seguridad.

- Recurso: es aquel que debe ser asegurado.
- Guardias de la política: los mecanismos que aseguran estos recursos.
- Políticas: documentos auto-procesables que describen las restricciones de estos recursos.

Las políticas se pueden dividir en:

- Políticas de permisos: se encargan de las acciones y accesos que las entidades **tienen permiso** de realizar.
- Políticas obligatorias: se encargan de las acciones y accesos que las entidades **deben** realizar.

La arquitectura se preocupa además de los guardias y su rol dentro de la arquitectura. Por ejemplo, el acceso no autorizado puede ser controlado por los guardias, que identifican los agentes que desean acceder al recurso. Este mecanismo deberá ser controlado por el documento de políticas, que indica que información debe ser proporcionada para permitir el acceso. Una política de permisos habilita o deshabilita el acceso a un recurso o acción. Por ejemplo, en

SOAP, los intermediarios juegan un papel importante, pues estos no permitirán que el mensaje sea transmitido si alguna política de seguridad ha sido violada.

Sin embargo, no todos los guardias son procesos. Para precautelar la información también se encriptan los mensajes, tanto su contenido, como identificadores de los agentes solicitante y receptor, siendo, en este caso, el guardia la encriptación en sí.

4.6.2 Amenazas de seguridad a nivel de mensajes

Los mecanismos a nivel de red tradicionales como SSL, VPNs, IPSec y MIME son punto a punto. Aunque estas tecnologías se usan en la seguridad de los servicios web, estos se encuentran en la capa de aplicación del modelo OSI y, por tanto, estos mecanismos no son suficientes para una seguridad en un contexto end-to-end, el cual necesita de mayor granularidad. La Figura 4-8 ilustra esta situación, en donde, debemos tomar en cuenta que los servicios web usan un enfoque basado en mensajes, que realizan interacciones complejas, pudiendo incluir el ruteo de los mensajes **entre varios dominios seguros**, hasta llegar a su destino. Razón por la cual, es necesario incluir seguridad a nivel de los mensajes transmitidos.

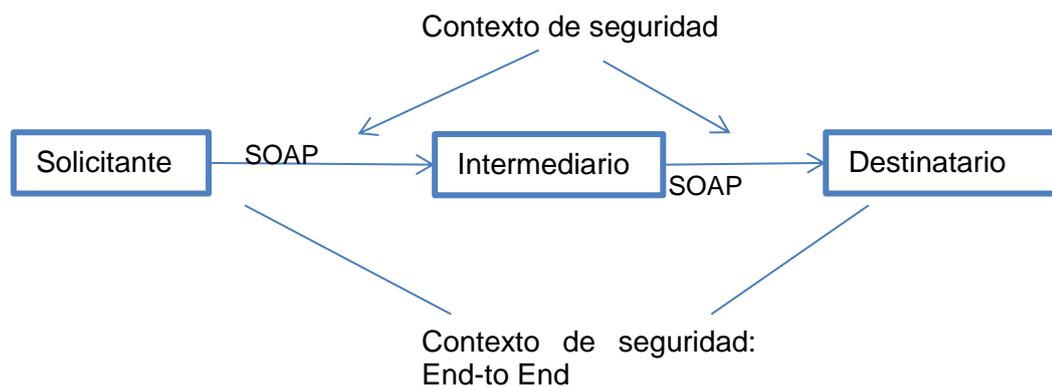


Figura 4-8: Seguridad End-to-End

Las posibles amenazas dirigidas a la seguridad de los mensajes son las siguientes:

- Alteración de mensajes: estas amenazas afectan la integridad de los mensajes, modificando partes o todo el mensaje. También se puede afectar la integridad de los mensajes manipulando sus adjuntos.
- Confidencialidad: Entidades no autorizadas acceden a información del mensaje o partes del mismo. Por ejemplo, información de una tarjeta de crédito.
- Man-in-the-middle: El agresor ataca a un intermediario e intercepta los mensajes entre el solicitante y destinatario. El agresor actuará de tal manera que los implicados pensarán que se están comunicando entre sí.
- Suplantación (Spoofing): El agresor asume la identidad de una entidad de confianza para sabotear la seguridad de la entidad objetivo. La entidad objetivo pensará que se está comunicando con una entidad de confianza.
- Negación del servicio (DoS): el ataque se enfoca en evitar que los usuarios puedan hacer uso de los servicios. Esto se realiza mediante la interceptación de mensajes.
- Replica de ataques: En este ataque, el agresor intercepta un mensaje y lo replica al agente que desea atacar, para saturarlo. Esto puede ser evitado mediante técnicas de autenticación, junto con tiempos de expiración y numeración secuencial de mensajes.

4.6.3 Requerimientos de seguridad

El objetivo de la seguridad en un web service es crear un ambiente, en donde los mensajes puedan ser conducidos desde el origen a su destino de una manera segura, sin importar si existen o no intermediarios.

Los requerimientos para lograr seguridad entre el solicitante y proveedor del servicio son:

4.6.3.1 Mecanismos de Autenticación

Son necesarios para verificar la identidad de los agentes de ambas partes. En ocasiones, es necesario el uso de autenticación mutua, pues los participantes no necesariamente se encontraran conectados al mismo intermediario.

Existen muchas formas de autenticación: contraseñas, pases únicos y certificados. Cuando se utilicen contraseñas, estas deben ser seguras y se debe implementar otro tipo de seguridad de autorización como: certificados, Lightweight Directory Access Protocol (LDAP protocolo ligero de acceso a directorios), Remote Authentication Dial-in User Service (RADIUS), Kerberos o Public Key Infrastructure (PKI Infraestructura de llave pública).

4.6.3.2 Autorización

Es necesaria para controlar acceso a los recursos. Una vez realizada la autenticación, estos mecanismos controlan que el solicitante acceda a los recursos apropiados. La política determina los derechos del solicitante.

4.6.3.3 Integridad y confidencialidad de los datos

Las técnicas de integridad de los datos se aseguran de que los datos no hayan sido alterados o modificados en una transmisión sin ser detectados. La confidencialidad, por su parte, se encarga de que los datos sean accedidos únicamente por las partes autorizadas. Las técnicas de encriptación y firma digital también se pueden usar para estos propósitos.

4.6.3.4 Integridad de las transacciones y comunicaciones

Es necesaria para asegurar que los procesos del negocio fueron realizados de una manera apropiada y el flujo de operaciones se realizó de una manera correcta.

4.6.3.5 No Repudio

Es un servicio de seguridad que protege a una parte contra la falsa negación de la llegada de un mensaje. Estas tecnologías proporcionan evidencia de la llegada de los mensajes, que puede usarse para resolver dicho conflicto.

4.6.3.6 Integridad y confidencialidad de mensajes emisor-receptor

La integridad y confidencialidad de mensajes debe ser asegurada incluso con la presencia de intermediarios.

4.6.3.7 Pistas de auditoría

Son necesarias para rastrear los accesos de los usuarios y su comportamiento, así como, para verificar la integridad del sistema. Los agentes son los encargados de ejecutarlas, y monitorear: los recursos y otros agentes, tomando en cuenta que las obligaciones establecidas sean respetadas y dadas de alta. Si un agente detecta la violación de una política, algún tipo de acción debe ser promulgada.

4.6.3.8 Ejecución distribuida de políticas de seguridad

Se debe definir políticas de seguridad y refuerzo en las diferentes plataformas con privilegios variados.

4.6.4 Consideraciones de seguridad de la arquitectura

Las organizaciones que hagan uso de servicios web deben hacerlo de una manera segura, incluyendo el ruteo, gestión, publicación y descubrimiento. Se deben considerar el uso de servicios de seguridad como autenticación, autorización, encriptación y auditoría.

Además, se debe tomar en cuenta que los servicios web pueden pasar por firewalls, y pueden ser canalizados por puertos y protocolos existentes. Por lo que, es importante la implementación de las siguientes capacidades:

- Identidad inter-dominio: es importante para los servicios web el soportar el mapeo de identidades entre diferentes dominios, o incluso dentro del mismo. Una entidad proveedora y una solicitante pueden usar sus identidades para encriptar y firmar los mensajes que intercambiarán.
- Políticas distribuidas: son políticas de seguridad asociadas con el solicitante, el servicio y el mecanismo de descubrimiento del servicio. Estas políticas se utilizan para definir los privilegios de acceso en la solicitud y los mensajes intercambiados. Cada parte es la encargada de validar sus políticas.
- Políticas de confianza: son políticas distribuidas que se aplican al ambiente de la contraparte. El solicitante debe confiar en el ambiente de un servicio y, el proveedor, en la entidad solicitante.
- Mecanismo de descubrimiento seguro: refuerza las políticas para la publicación y descubrimiento de un servicio.
- Confianza y descubrimiento: se refiere principalmente al caso en que una entidad descubre un nuevo servicio, del que no era consciente, y, por lo tanto, debe poder decidir si confiar o no en el servicio de esta entidad desconocida. Resaltando que el descubrimiento puede ser manual (juzgamiento humano) o automático.
- Mensajería segura: asegura privacidad, confidencialidad e integridad en las interacciones. Ejemplos de estas técnicas, como la encriptación y la firma de mensajes, pueden ser utilizadas en el ruteo y mensajería confiable.
- Consideraciones de privacidad: la privacidad relacionada con el comportamiento, hábitos y acciones está expresada en términos de políticas

que tienen los propietarios, junto con los mecanismos para asegurar que sus derechos son respetados. Las políticas de privacidad son más obligatorias que las de control de acceso.

4.6.5 Desafíos a la seguridad de los servicios Web³⁴

En la actualidad existen recomendaciones para abordar la seguridad de los web services a través de marcos de trabajo definidos por OASIS. OASIS define WS-security (seguridad en Web services) para tratar las siguientes consideraciones:

- Las relaciones de confianza entre las organizaciones podrían necesitar la definición de contratos legales y responsabilidades.
- Los requerimientos a un Web Service deberían ser autenticados y autorizados apropiadamente.
- Los mensajes hacia y desde un Web Service deberían ser protegidos de accesos y modificaciones no autorizadas.

WS-Security no es un reemplazo de alguna tecnología de seguridad, sino que provee un modelo unificado que utiliza las tecnologías existentes, y que permite que las aplicaciones intercambien mensajes SOAP de forma segura.

WS-I es una organización que define cómo se deberían cumplir los requerimientos de interoperabilidad, al utilizar un conjunto de tecnologías que aseguran la transmisión de mensajes SOAP. Según la WS-I los desafíos de la seguridad en la transmisión de mensajes SOAP son los siguientes:

- Identificación y Autenticación de las partes.
- Identificación y Autenticación de los datos de origen.
- Integridad de los datos: Integridad en el transporte y del mensaje SOAP.

³⁴ Sección tomada de Seguridad en Web Services (Seguel)

- Confidencialidad de los datos: Confidencialidad en el transporte y del mensaje SOAP.
- Unicidad del mensaje SOAP.

Los estándares que permiten abordar estos desafíos son los siguientes:

- WS-Security (Web Services Security): provee un marco de trabajo para el transporte seguro de mensajes SOAP.
- WS-Trust: describe un modelo conceptual de confianza que permite que los Web Services puedan interoperar en forma segura.
- WS-Policy: describe un marco de trabajo para definir permisos y restricciones de las políticas de seguridad entre los participantes de la transmisión de los mensajes SOAP.
- WS-SecureConversation: define un marco de trabajo para manejar y autenticar el intercambio de mensajes SOAP, incluyendo su contexto de seguridad.
- WS-Federation: define cómo establecer relaciones de confianza entre dominios de seguridad.
- WS-Privacy: describe la sintaxis y semántica para la comunicación privada de políticas de seguridad para servicios web e instancias de datos en mensajes.
- WS-Authorization: describe cómo las políticas de acceso para un Web Service son especificadas y gestionadas.
- SAML (Security Assertion Markup Language): provee un marco de trabajo para intercambiar información en forma segura.
- XACML (eXtensible Access Control Markup Language): es un lenguaje para escribir políticas de control de acceso.

- XKMS (XML Key Management Specification): es un mecanismo basado en XML para la gestión de una Infraestructura de Clave Pública (PKI).
- XML Encryption: es un mecanismo para encriptar partes de un documento XML.
- XML Signature: es un mecanismo para firmar partes de un documento XML.

El uso de una arquitectura orientada a servicios SOA, junto con servicios web, es de gran ayuda para el intercambio de información entre organizaciones o para la creación de un middleware para el manejo de varias aplicaciones. En ambos casos, el uso de servicios web debe realizarse con estricto cuidado de los problemas de seguridad antes mencionados.

En el sistema propuesto, se ha optado por el uso de servicios web con tecnologías SOAP y WSDL, para que funcionen como un sistema de integración entre las aplicaciones de la Universidad del Azuay, es decir, como una plataforma Middleware al que pueden acceder distintas aplicaciones a ser desarrolladas a futuro. El uso de servicios web también facilitará, a los kioscos remotos, el acceso a los datos requeridos para otorgar los certificados.

CAPÍTULO V

5. ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE (ERS)

5.1 Introducción

El presente capítulo corresponde a una Especificación de Requisitos de Software del sistema para kioscos emisores de certificados académicos para estudiantes. Su contenido se ha elaborado de acuerdo al manual de procesos de certificados estándar realizado por la consultora “DYNAMIX SOLUTIONS CONSULTORES INTEGRALES.” En este capítulo, se describirá las funcionalidades que debe prestar el sistema para el cumplimiento de los requisitos del usuario (requerimientos funcionales) y de las restricciones del sistema (requerimientos no funcionales).

5.1.1 Propósito

El objetivo de la presente especificación es el definir de manera clara y precisa los requisitos, necesidades, funcionalidades y limitaciones del sistema propuesto. Constituyendo de esta manera, junto con los diagramas de casos de uso y de actividad, el análisis del sistema propuesto. Este documento va dirigido a autoridades de la Universidad del Azuay y constituirá el medio de comunicación entre las partes involucradas en desarrollo del sistema. El área de sistemas será la responsable de la validación del ERS para evitar inconvenientes en el desarrollo y facilitar posteriores modificaciones al sistema.

5.1.2 Ámbito del Sistema

El proyecto abarca desde la creación de los servicios web para la obtención de datos del estudiante, hasta la interfaz gráfica del kiosco que permita al estudiante identificarse, solicitar el certificado y finalmente recibirlo.

El sistema contempla lo siguiente:

- Identificación del estudiante.
- Elección del tipo de certificado.
- Generación de la Deuda.
- Impresión del Certificado.

El objetivo que se pretende con el sistema propuesto es el de disminuir los tiempos de espera, brindando así un mejor servicio a los estudiantes de la Universidad del Azuay.

5.1.3 Definiciones, Acrónimos y Abreviaturas

Dynamix Solutions DSCI	DYNAMIX SOLUTIONS CONSULTORES INTEGRALES, empresa de consultoría contratada por la Universidad del Azuay para estudio y mejora de procesos.
Kiosco	Computador especialmente diseñado para que mediante su sistema informático permita la impresión de certificados
Certificado	Es un tipo de texto administrativo empleado para constatar un hecho determinado.

5.1.4 Referencias

Como base para la realización de este documento se tiene como referencia los documentos realizados por la consultora DSCI:

- Manual de Procesos- Certificaciones estándar.
- Diagrama de Procesos- Certificaciones estándar.

5.1.5 Historial de Versiones

FECHA	VERSIÓN	CAMBIOS
12/10/2011	1.0	Versión Original del Documento
23/10/2011	1.1	Precisión en casos de uso de manejo de

		Información a mostrar.
15/02/2012	1.5	Se precisan de mejor manera los casos de uso, otorgando mayor organización y se reorganiza el diagrama de casos de uso.
29/02/2012	2.0	Se realizan precisiones para que el documento conste como un Capitulo de la tesis y se realiza una descripción extendida de los casos de uso.
03/04/2012	2.1	Se aumenta el certificado de Notas del ciclo actual

5.2 Descripción general

5.2.1 Perspectiva del Producto

El sistema deberá permitir mejorar la atención brindada por la Universidad del Azuay hacia los estudiantes, otorgándoles sus certificados de una manera ágil y oportuna. Liberando, además, parte de la carga de la realización de certificados a las secretarías de cada facultad.

5.2.1.1 Dependencias del producto.

- Interactuará con el Sistema de Créditos y el Sistema de Tesorería de la Universidad del Azuay.
- Dependerá de los usuarios y contraseñas del área de internet.

5.2.1.2 Interfaces hardware

- Constará de un kiosco con las siguientes partes esenciales:
 - Pantalla táctil.
 - Impresora.
 - Interfaz de comunicación.

5.2.1.3 Interfaces de Software

El sistema presentará una interfaz que facilite su uso mediante la pantalla táctil, de acuerdo a los lineamientos para interfaces gráficas de pantallas táctiles. Esta interfaz, deberá seguir los estándares en cuanto al diseño de interfaces web dentro de la Universidad del Azuay.

5.2.2 Funciones del Producto

El sistema permite a los estudiantes de la Universidad del Azuay obtener certificados académicos de manera fácil y rápida; manteniendo la misma validez que los entregados actualmente.

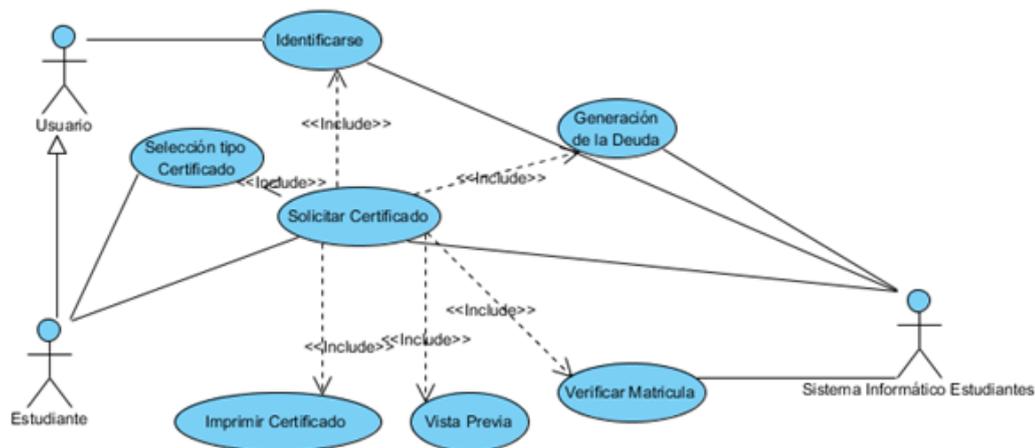


Figura 5-1: Diagrama de Casos de uso esencial

5.2.3 Características de los usuarios

Los usuarios del sistema serán estudiantes universitarios, por lo que, se asume que deben tener un conocimiento básico del manejo de un computador. Sin embargo, el sistema debe ser lo más intuitivo posible y su interfaz será similar al manejado por los estudiantes para realizar sus trámites por Internet, lo que permitirá una mayor facilidad de uso, sin entrenamiento alguno.

5.2.4 Restricciones

El sistema podrá emitir certificados solamente para los estudiantes pertenecientes al sistema de créditos (código >60000), que actualmente se encuentran matriculadas en la Universidad del Azuay o estén en condición de egresados, pues solamente a estas personas se les podrá cargar el valor del certificado como deuda pendiente.

Se mantendrá una interfaz de usuario acorde a los colores que identifican a la Universidad del Azuay, siguiendo la línea de diseño que esta maneja en las aplicaciones de acceso estudiantil.

5.2.5 Suposiciones y Dependencias

- Para que el sistema sea completamente funcional, se asume que la Universidad cuenta con el presupuesto necesario para la adquisición de los Kioscos y tendrá asignado los espacios correspondientes para la instalación de los mismos.
- Así también, se supone que los requisitos descritos en la presente especificación no serán modificados después de que sean aprobados por las autoridades de la Universidad.

5.2.6 Requisitos Futuros

Como complemento al trabajo que se va a realizar, posteriormente se deberá complementar el sistema, para permitir la emisión los certificados a los estudiantes antiguos (código <60000) de la Universidad, así como la inclusión de más certificados dentro del servicio que presente el Kiosco.

5.3 Requisitos Específicos

5.3.1 Requisitos Funcionales

Los requisitos funcionales serán agrupados mediante los casos de uso de la Figura 5-2, estos permiten: dar una perspectiva del sistema desde el punto de vista de los usuarios o actores, y dar una visión general de los participantes y del alcance del sistema.

5.3.1.1 Descripción de Actores

- **Usuario:** Persona que no se ha identificado dentro del sistema, puede ser un estudiante de la Universidad o no.
- **Estudiante:** Usuario que se ha identificado y, por tanto, puede hacer uso del sistema para obtener certificados académicos.
- **Sistema Créditos Estudiantes:** Representa al sistema y base de datos externa que maneja los datos de los estudiantes, sus materias, calificaciones, etc.
- **Sistema de Tesorería:** Representa al sistema y base de datos de la Universidad, el cual, controla las deudas de los estudiantes y el cobro de las mismas.
- **Sistema Usuarios:** es el sistema y base de datos que contiene los usuarios y contraseñas de los estudiantes, los cuales se utilizan en los servicios prestados por Internet.

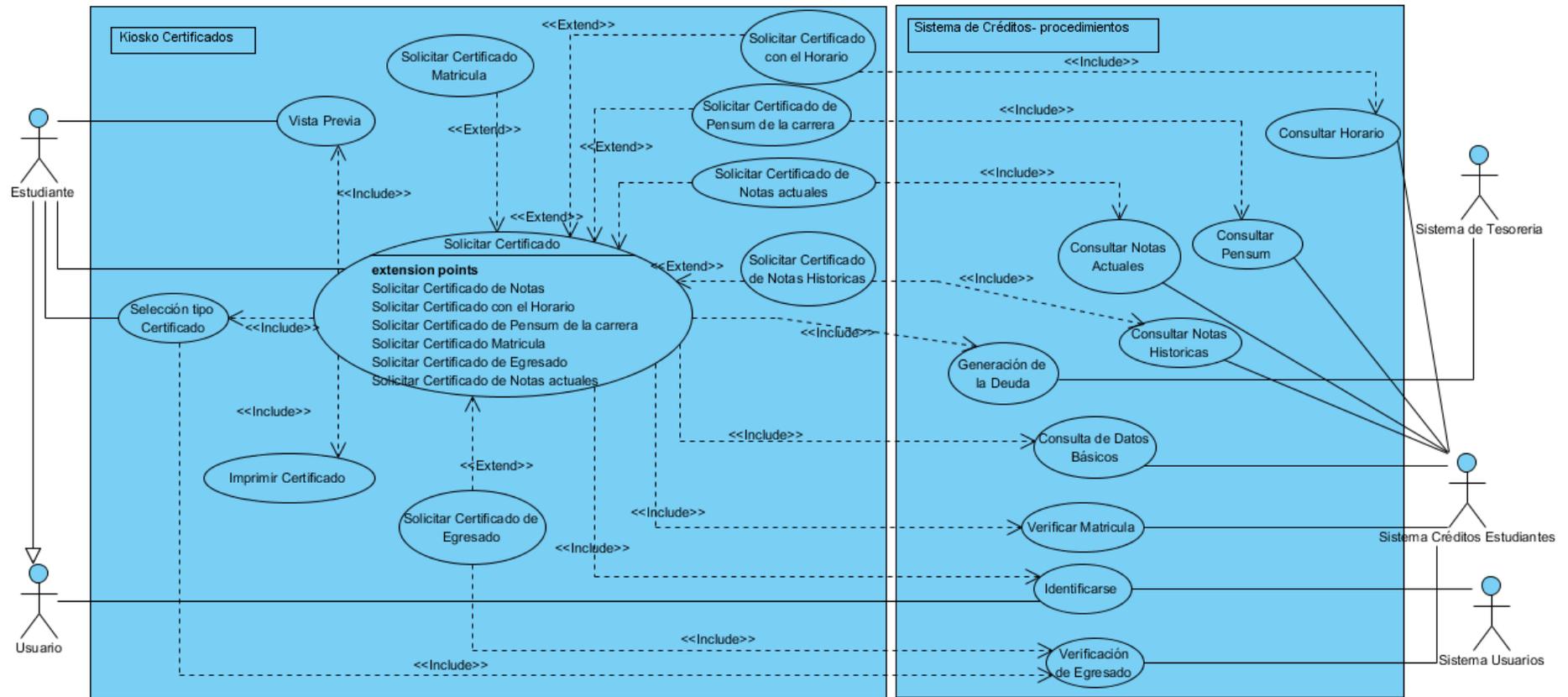


Figura 5-2: Diagrama de Casos de uso extendido

5.3.1.2 Descripción de Casos de uso

Diagrama de Actividad

El flujo de eventos descrito en cada caso de uso se basa en el diagrama de actividad de la Figura 5-3, el cual, recopila todo el proceso para la emisión de un certificado académico, sin tomar en cuenta el tipo de certificado que se solicite.

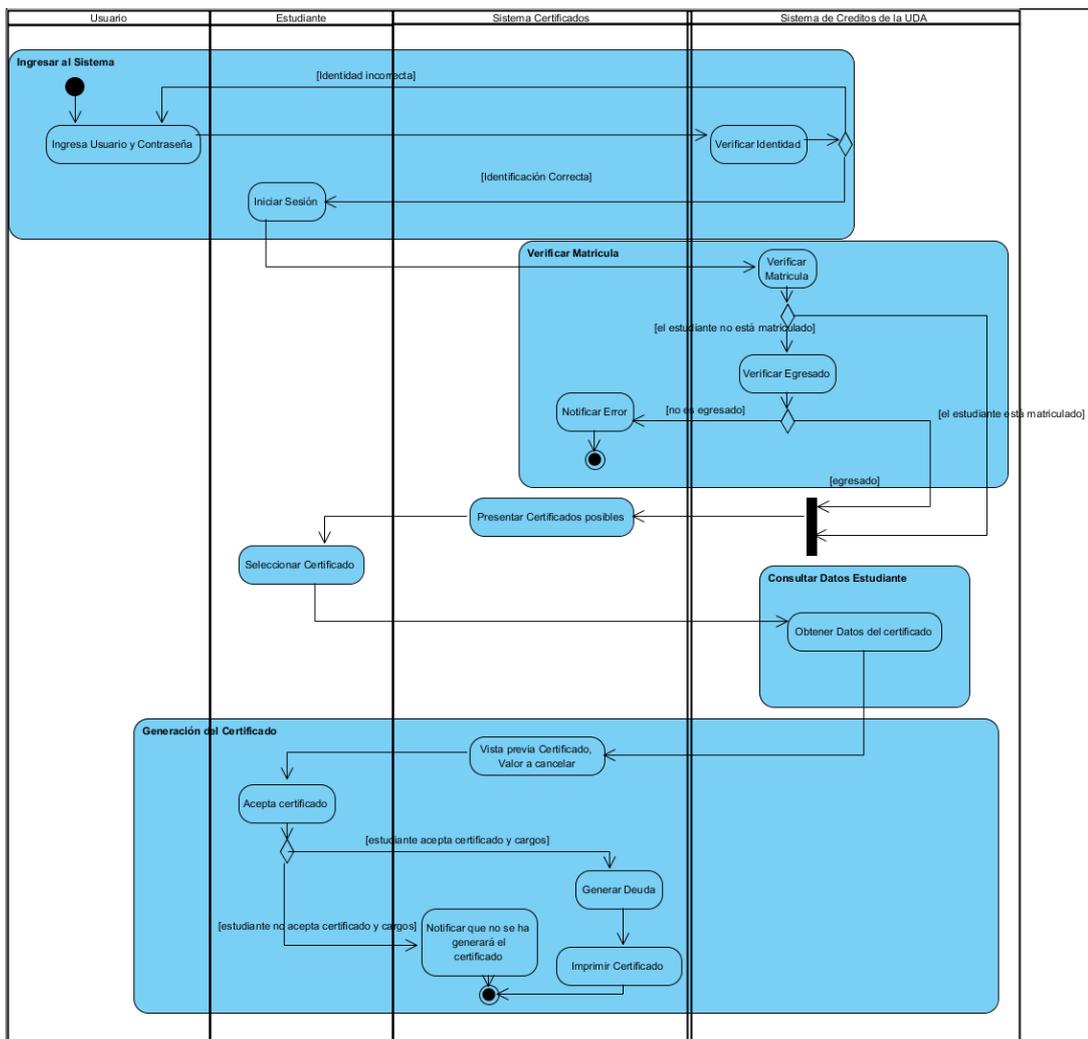


Figura 5-3: Diagrama de Actividad

Las prioridades de los casos de uso pueden ser:

Existente	El caso de uso es una extensión de otro sistema ya implementado
Opcional	Se especificará explícitamente por parte de la persona responsable si se automatizará las funcionalidades, bajo la supervisión del responsable del control de aceptación de los ERS.
Deseable	El responsable del control de aceptación de los ERS definirá la automatización de estas funcionalidades
Obligatorio	Estas funcionalidades serán automatizadas 100%

Caso de uso 1	Identificarse
Actores	Usuario Sistema Usuarios
Descripción	Se autenticará la identidad del usuario mediante su usuario y contraseña.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El Usuario toca la pantalla para comenzar la interacción. 2. El sistema presenta los campos de usuario y contraseña 3. El usuario rellena los campos de usuario y contraseña
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 3a. El usuario no completa la información. 3b. El usuario ingreso un usuario o contraseña invalido. 4. El sistema indica el error. 5. El usuario regresa al paso 2.
Pre-condición	El sistema se debe encontrar en estado de Espera
Post-condición	El estudiante se encontrará autenticado
Caso de éxito	Se mostrarán los certificados que el Estudiante puede solicitar.
Triggers	El Usuario toca la pantalla para comenzar la interacción.
Requerimientos Funcionales	<p>R.1.1 El sistema verificará la identidad del usuario.</p> <p>R.1.2 El sistema utilizará los mismos usuarios y contraseñas que para el resto de servicios de la Universidad.</p> <p>R.1.3 El sistema indicará si se ha ingresado erróneamente el usuario y/o la contraseña.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p>
Requerimientos Especiales	

Caso de uso 2	Verificar Matricula
Actores	Sistema Crédito Estudiantes
Descripción	Se verificará que el estudiante se encuentre matriculado en la Universidad al momento de seleccionar el certificado.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema consultará en la base de datos el estado del estudiante. 2. El sistema indicará el estado del estudiante.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. El sistema encuentra un error al consultar la base de datos. 2a. El sistema reportará un error.
Pre-condición	El sistema debe haber identificado al estudiante
Post-condición	El sistema pasará a modo de espera de la selección del certificado
Caso de éxito	Se mostrarán los certificados que se pueden emitir
Triggers	Solicitud de verificación por medio de web service, luego de la identificación.
Requerimientos Funcionales	<p>R.2.1 El sistema verificará que el usuario sea un estudiante matriculado actualmente.</p> <p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p> <p>R.2.4 El sistema emitirá un certificado de que un estudiante se encuentra matriculado.</p>
Requerimientos Especiales	

Caso de uso 3	Verificación de Egresados
Actores	Sistema Créditos Estudiantes
Descripción	Se verificará si el estudiante se encuentra egresado en caso de no estar matriculado.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema consultará en la base de datos el estado del estudiante. 2. El sistema indicará el estado del estudiante.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. El sistema encuentra un error al consultar la base de datos. 2a. El sistema reportará error.
Pre-condición	El estudiante no se encuentra matriculado
Post-condición	El sistema pasará a modo de espera de la selección del certificado
Caso de éxito	Se mostrarán los certificados que se pueden emitir.
Triggers	Solicitud de verificación por medio de web service, luego de la verificación de matrícula.

Requerimientos Funcionales	<p>R.3.1 El sistema verificará que el usuario tenga el estado de egresado.</p> <p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p> <p>R.3.2 El sistema emitirá el certificado de egresado únicamente a estudiantes con estado de egresado en el sistema de créditos.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p> <p>R.3.3 El sistema emitirá un certificado que indique que un estudiante este egresado.</p>
Requerimientos Especiales	

Caso de uso 4	Consulta de Datos Básicos
Actores	Sistema Créditos Estudiantes
Descripción	Consultará los datos básicos del Estudiante, tales como nombres, apellidos, carrera, facultad.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema consultará en la base de datos, los datos básicos del estudiante. 2. El sistema devolverá los datos del estudiante.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. El sistema encuentra un error al consultar la base de datos. 2a. El sistema reportará error.
Pre-condición	El sistema ha identificado al estudiante
Post-condición	El sistema se encontrará esperando la selección del certificado
Caso de éxito	Se contará con la información básica del estudiante
Triggers	Solicitud de verificación por medio de web service luego de la identificación del estudiante
Requerimientos Funcionales	<p>R.4.1 El sistema mostrará en el certificado los nombres y apellidos del estudiante.</p> <p>R.4.2 El sistema indicará en el certificado el número de cédula del estudiante.</p> <p>R.4.3 El sistema mostrará en el certificado la carrera y facultad del estudiante.</p>
Requerimientos Especiales	

Caso de uso 5	Consultar Pensum
Actores	Sistema Créditos Estudiantes
Descripción	Consultará la malla curricular del pensum y carrera que el estudiante este cursando.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema consultará en la base de datos el pensum del estudiante. 2. El sistema devolverá el pensum del estudiante.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. El sistema encuentra un error al consultar la base de datos. 2a. El sistema reportará error.
Pre-condición	
Post-condición	
Caso de éxito	Se obtendrá la información del pensum del estudiante
Triggers	Solicitud de verificación por medio de web service al seleccionar certificado de pensum.
Requerimientos Funcionales	<p>R.5.1 El sistema emitirá un certificado con el pensum académico del estudiante.</p> <p>R.5.2 El sistema ordenará el certificado de pensum por ciclos.</p> <p>R.5.3 El sistema mostrará el código, nombre, número de créditos y prerrequisitos de cada materia en el certificado de pensum.</p>
Requerimientos Especiales	

Caso de uso 6	Consultar Notas Históricas
Actores	Sistema Créditos Estudiantes
Descripción	Consultará las notas históricas del estudiante, organizadas mediante el pensum de la carrera que el estudiante este cursando.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema consultará en la base de datos las calificaciones del estudiante. 2. El sistema devolverá las calificaciones históricas del estudiante.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. El sistema encuentra un error al consultar la base de datos. 2a. El sistema reportará error.
Pre-condición	
Post-condición	
Caso de éxito	Se obtendrá la información de las calificaciones históricas del estudiante
Triggers	Solicitud de verificación por medio de web service al seleccionar certificado de notas históricas.
Requerimientos Funcionales	R.6.1 El sistema emitirá un certificado con las notas históricas del estudiante.

	<p>R.6.2 El sistema ordenará el certificado de notas históricas por ciclos.</p> <p>R.6.3 El sistema indicará si un estudiante aprobó una materia en primera, segunda o tercera matrícula en el certificado de notas históricas.</p> <p>R.6.4 El sistema indicará si la materia ha sido aprobada o reprobada en el certificado de notas.</p> <p>R.6.5 El sistema mostrará el código, nombre y número de créditos de cada materia en el certificado de notas.</p>
Requerimientos Especiales	

Caso de uso 6a	Consultar Notas Actuales
Actores	Sistema Créditos Estudiantes
Descripción	Consultará las notas del estudiante en el periodo actual.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema consultará en la base de datos las calificaciones del estudiante en el periodo actual. 2. El sistema devolverá las calificaciones del estudiante.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. El sistema encuentra un error al consultar la base de datos. 2a. El sistema reportará error.
Pre-condición	
Post-condición	
Caso de éxito	Se obtendrá la información de las calificaciones del estudiante en el periodo actual.
Triggers	Solicitud de verificación por medio de web service al seleccionar certificado de notas actuales.
Requerimientos Funcionales	<p>R.6.4 El sistema indicará si la materia ha sido aprobada o reprobada en el certificado de notas.</p> <p>R.6.5 El sistema mostrará el código, nombre y número de créditos de cada materia en el certificado de notas.</p> <p>R.6.6 El sistema indicará en el certificado de notas actuales, las notas del periodo actual, correspondientes al aporte, examen, suspenso y nota final.</p> <p>R.6.7 El sistema emitirá un certificado de las notas del estudiante en el periodo actual.</p>
Requerimientos Especiales	

Caso de uso 7	Consultar Horario
Actores	Sistema Créditos Estudiantes
Descripción	Consultará el horario del estudiante en el ciclo en el que se encuentra matriculado.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema consultará en la base de datos el horario actual del estudiante. 2. El sistema devolverá el horario organizado del estudiante.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. El sistema encuentra un error al consultar la base de datos. 2a. El sistema reportará error.
Pre-condición	
Post-condición	
Caso de éxito	Se obtendrá la información del horario del estudiante
Triggers	Solicitud de verificación por medio de web service al seleccionar certificado con el horario.
Requerimientos Funcionales	R.7.1 El sistema emitirá un certificado con el horario actual del estudiante.
Requerimientos Especiales	

Caso de uso 8	Generación de la Deuda
Actores	Sistema de tesorería
Descripción	Se acumulará el valor total del certificado a la cuenta del estudiante.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. Se aceptan los cargos por la emisión del certificado 2. El sistema ingresará a la tabla correspondiente e ingresará un nuevo registro con la deuda del estudiante. 3. El sistema indicará si la deuda ha sido guardada.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 2a. El sistema encuentra un error al ingresar en la base de datos. 3a. El sistema reportará error al guardar los datos.
Pre-condición	El sistema debe encontrarse en la vista previa del certificado
Post-condición	El sistema deberá imprimir el certificado.
Caso de éxito	Se imprimirá el certificado
Triggers	Aceptación de los cargos y la deuda al mostrar la vista previa
Requerimientos Funcionales	<p>R.8.1 El sistema generará la deuda correspondiente al certificado, en la cuenta del estudiante.</p> <p>R.8.2 El sistema preguntará al estudiante si está de acuerdo con el valor que será acumulado en su cuenta por la emisión del certificado.</p> <p>R.8.3 El sistema notificará al estudiante el valor que será</p>

	<p>acumulado en su cuenta con la Universidad.</p> <p>R.8.4 El sistema notificará al estudiante por correo electrónico que se ha generado la deuda en su cuenta.</p> <p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p>
Requerimientos Especiales	

Caso de uso 9	Selección tipo Certificado
Actores	Estudiante
Descripción	Permitirá seleccionar el tipo de certificado a emitir
Flujo de Eventos Principal	1. El estudiante selecciona el tipo de certificado que desea obtener.
Flujo de Eventos Alternativo	
Pre-condición	El sistema se debe encontrar en “esperando selección”, luego de mostrar los certificados que se puede emitir a ese estudiante.
Post-condición	El sistema debe mostrar la vista previa del certificado solicitado.
Caso de éxito	Se mostrará la vista previa del certificado solicitado.
Triggers	Al seleccionar un tipo de certificado de las opciones presentadas
Requerimientos Funcionales	<p>R.2.4 El sistema emitirá un certificado de que un estudiante se encuentra matriculado.</p> <p>R.3.3 El sistema emitirá un certificado que indique que un estudiante este egresado.</p> <p>R.5.1 El sistema emitirá un certificado con el pensum académico del estudiante.</p> <p>R.6.1 El sistema emitirá un certificado con las notas históricas del estudiante.</p> <p>R.6.7 El sistema emitirá un certificado de las notas del estudiante en el periodo actual.</p> <p>R.7.1 El sistema emitirá un certificado con el horario actual del estudiante.</p> <p>R.9.1 El sistema permitirá seleccionar el certificado a emitir entre los certificados disponibles para ese estudiante.</p> <p>R.3.2 El sistema emitirá el certificado de egresado únicamente a estudiantes con estado de egresado en el sistema de créditos.</p>
Requerimientos Especiales	

Caso de uso 10	Vista Previa
Actores	Estudiante
Descripción	Mostrará en pantalla el certificado a emitir.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. Se selecciona el tipo de certificado 2. Se mostrará una vista previa del certificado. 3. Se consultará si el estudiante acepta los cargos y desea imprimir el certificado.
Flujo de Eventos Alternativo	
Pre-condición	El estudiante ha seleccionado el certificado a emitir
Post-condición	El estudiante observa el certificado a obtener
Caso de éxito	Se mostrará la vista previa del certificado a emitir.
Triggers	Al seleccionar el tipo de certificado.
Requerimientos Funcionales	<p>R.10.1 El sistema presentará el certificado a obtener previo a su impresión.</p> <p>R.8.2 El sistema preguntará al estudiante si está de acuerdo con el valor que será acumulado en su cuenta por la emisión del certificado.</p>
Requerimientos Especiales	

Caso de uso 11	Imprimir Certificado
Actores	Estudiante
Descripción	Se emitirá el certificado solicitado por el estudiante.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El estudiante acepta los cargos y la emisión del certificado. 2. El certificado se imprime.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. El estudiante no acepta los cargos. 2a. El certificado no se emite.
Pre-condición	Se ha presentado la vista previa del certificado
Post-condición	El sistema regresa el estado de espera
Caso de éxito	El certificado se emite.
Triggers	Al aceptar los cargos y la impresión del certificado
Requerimientos Funcionales	<p>R.11.1 El sistema imprimirá el certificado seleccionado por el usuario.</p> <p>R.11.2 El sistema garantizará la validez de los certificados que emita.</p> <p>R.11.3 El sistema colocará la firma digitalizada del/de la secretario(a) de la facultad a la que corresponde el estudiante.</p>
Requerimientos Especiales	

Caso de uso 12	Solicita Certificado
Actores	Estudiante
Descripción	Se emitirá el certificado solicitado por el estudiante, previa verificación de su identidad y generación de la deuda.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El estudiante selecciona el certificado que desea solicitar. 2. El sistema consulta los datos necesarios para ser presentados en el certificado, mediante web service. 3. El sistema muestra una vista previa del certificado y el valor que se cargará a la cuenta del estudiante por la emisión. 4. El estudiante acepta los cargos por la emisión del certificado. 5. El sistema genera los cargos en la cuenta del estudiante. 6. El sistema Imprime el certificado. 7. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Flujo de Eventos Alternativo	<p>2a. Se produce un error al consultar los datos</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>4a. El estudiante no acepta los cargos por la emisión del certificado.</p> <ol style="list-style-type: none"> 1. El sistema regresa a la selección del certificado. <p>5a. Se presenta un error al generar los cargos.</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Pre-condición	El estudiante debe identificarse
Post-condición	El sistema regresará al estado de espera
Caso de éxito	Se emitirá el certificado solicitado
Triggers	Al seleccionar un tipo de certificado.
Requerimientos Funcionales	<p>R.1.1 El sistema verificará la identidad del usuario.</p> <p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p> <p>R.2.4 El sistema emitirá un certificado de que un estudiante se encuentra matriculado.</p> <p>R.3.3 El sistema emitirá un certificado que indique que un estudiante este egresado.</p> <p>R.5.1 El sistema emitirá un certificado con el pensum académico del estudiante.</p> <p>R.6.1 El sistema emitirá un certificado con las notas históricas del estudiante.</p>

	<p>R.6.7 El sistema emitirá un certificado de las notas del estudiante en el periodo actual.</p> <p>R.7.1 El sistema emitirá un certificado con el horario actual del estudiante.</p> <p>R.10.1 El sistema presentará el certificado a obtener previo a su impresión.</p> <p>R.8.2 El sistema preguntará al estudiante si está de acuerdo con el valor que será acumulado en su cuenta por la emisión del certificado.</p> <p>R.8.1 El sistema generará la deuda correspondiente al certificado, en la cuenta del estudiante.</p> <p>R.8.4 El sistema notificará al estudiante por correo electrónico que se ha generado la deuda en su cuenta.</p> <p>R.11.1 El sistema imprimirá el certificado seleccionado por el usuario.</p> <p>R.11.3 El sistema colocará la firma digitalizada del/de la secretario(a) de la facultad a la que corresponde el estudiante.</p> <p>R.4.1 El sistema mostrará en el certificado los nombres y apellidos del estudiante.</p> <p>R.4.2 El sistema indicará en el certificado el número de cédula del estudiante.</p> <p>R.4.3 El sistema mostrará en el certificado la carrera y facultad del estudiante.</p>
Requerimientos Especiales	

Caso de uso 13	Solicita Certificado de Matricula
Actores	Estudiante
Descripción	Se emitirá el certificado indicando en que materias el estudiante se encuentra actualmente matriculado.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El estudiante selecciona el certificado de matrícula. 2. El sistema consulta los datos generales del estudiante para ser presentados en el certificado, mediante web service. 3. El sistema muestra una vista previa del certificado y el valor que se cargará a la cuenta del estudiante por la emisión. 4. El estudiante acepta los cargos por la emisión del certificado. 5. El sistema genera los cargos en la cuenta del estudiante. 6. El sistema Imprime el certificado. 7. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Flujo de Eventos	2a. Se produce un error al consultar los datos

Alternativo	<ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>4a. El estudiante no acepta los cargos por la emisión del certificado.</p> <ol style="list-style-type: none"> 1. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>5a. Se presenta un error al generar los cargos.</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Pre-condición	El estudiante debe identificarse
Post-condición	El sistema regresará al estado de espera
Caso de éxito	Se emitirá el certificado solicitado
Triggers	Al seleccionar el certificado de matrícula.
Requerimientos Funcionales	<p>R.1.1 El sistema verificará la identidad del usuario.</p> <p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p> <p>R.2.4 El sistema emitirá un certificado de que un estudiante se encuentra matriculado.</p> <p>R.10.1 El sistema presentará el certificado a obtener previo a su impresión.</p> <p>R.8.2 El sistema preguntará al estudiante si está de acuerdo con el valor que será acumulado en su cuenta por la emisión del certificado.</p> <p>R.8.1 El sistema generará la deuda correspondiente al certificado, en la cuenta del estudiante.</p> <p>R.8.4 El sistema notificará al estudiante por correo electrónico que se ha generado la deuda en su cuenta.</p> <p>R.11.1 El sistema imprimirá el certificado seleccionado por el usuario.</p> <p>R.11.3 El sistema colocará la firma digitalizada del/de la secretario(a) de la facultad a la que corresponde el estudiante.</p> <p>R.4.1 El sistema mostrará en el certificado los nombres y apellidos del estudiante.</p> <p>R.4.2 El sistema indicará en el certificado el número de cédula del estudiante.</p> <p>R.4.3 El sistema mostrará en el certificado la carrera y facultad del estudiante.</p> <p>R.13.1 El certificado de Matrícula deberá indicar las materias en las que el estudiante se encuentra matriculado.</p> <p>R.13.2 El certificado de Matrícula mostrará el porcentaje de asistencia de cada materia hasta la fecha de emisión del</p>

	<p>certificado.</p> <p>R.13.3 El certificado de Matrícula tendrá el formato del Anexo 1</p>
Requerimientos Especiales	

Caso de uso 14	Solicita Certificado de Egresado
Actores	Estudiante
Descripción	Se emitirá el certificado de que el estudiante tiene un estado de egresado.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El estudiante selecciona el certificado de egresado. 2. El sistema consulta los datos generales del estudiante para ser presentados en el certificado, mediante web service. 3. El sistema muestra una vista previa del certificado y el valor que se cargará a la cuenta del estudiante por la emisión. 4. El estudiante acepta los cargos por la emisión del certificado. 5. El sistema genera los cargos en la cuenta del estudiante. 6. El sistema Imprime el certificado. 7. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Flujo de Eventos Alternativo	<p>2a. Se produce un error al consultar los datos</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>4a. El estudiante no acepta los cargos por la emisión del certificado.</p> <ol style="list-style-type: none"> 1. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>5a. Se presenta un error al generar los cargos.</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Pre-condición	El estudiante debe identificarse y tener el estado de egresado
Post-condición	El sistema regresará al estado de espera
Caso de éxito	Se emitirá el certificado de egresado
Triggers	Al seleccionar un certificado de egresado.
Requerimientos	R.1.1 El sistema verificará la identidad del usuario.

Funcionales	<p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p> <p>R.3.2 El sistema emitirá el certificado de egresado únicamente a estudiantes con estado de egresado en el sistema de créditos.</p> <p>R.3.3 El sistema emitirá un certificado que indique que un estudiante este egresado.</p> <p>R.10.1 El sistema presentará el certificado a obtener previo a su impresión.</p> <p>R.8.2 El sistema preguntará al estudiante si está de acuerdo con el valor que será acumulado en su cuenta por la emisión del certificado.</p> <p>R.8.1 El sistema generará la deuda correspondiente al certificado, en la cuenta del estudiante.</p> <p>R.8.4 El sistema notificará al estudiante por correo electrónico que se ha generado la deuda en su cuenta.</p> <p>R.11.1 El sistema imprimirá el certificado seleccionado por el usuario.</p> <p>R.11.3 El sistema colocará la firma digitalizada del/de la secretario(a) de la facultad a la que corresponde el estudiante.</p> <p>R.4.1 El sistema mostrará en el certificado los nombres y apellidos del estudiante.</p> <p>R.4.2 El sistema indicará en el certificado el número de cédula del estudiante.</p> <p>R.4.3 El sistema mostrará en el certificado la carrera y facultad del estudiante.</p> <p>R.14.1 El certificado de Egresado tendrá el formato del Anexo 2.</p>
Requerimientos Especiales	

Caso de uso 15	Solicita Certificado con el Horario
Actores	Estudiante
Descripción	Se emitirá el certificado con el horario de clases del estudiante, en el ciclo actual.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El estudiante selecciona el certificado con el horario de clases. 2. El sistema consulta los datos del horario actual para ser presentados en el certificado, mediante web service. 3. El sistema muestra una vista previa del certificado y el valor que se cargará a la cuenta del estudiante

	<p>por la emisión.</p> <ol style="list-style-type: none"> 4. El estudiante acepta los cargos por la emisión del certificado. 5. El sistema genera los cargos en la cuenta del estudiante. 6. El sistema Imprime el certificado. 7. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Flujo de Eventos Alternativo	<p>2a. Se produce un error al consultar los datos del horario</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>4a. El estudiante no acepta los cargos por la emisión del certificado.</p> <ol style="list-style-type: none"> 1. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>5a. Se presenta un error al generar los cargos.</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Pre-condición	El estudiante debe identificarse
Post-condición	El sistema regresará al estado de espera
Caso de éxito	Se emitirá el certificado con el horario.
Triggers	Al seleccionar el certificado con el horario.
Requerimientos Funcionales	<p>R.1.1 El sistema verificará la identidad del usuario.</p> <p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p> <p>R.7.1 El sistema emitirá un certificado con el horario actual del estudiante.</p> <p>R.10.1 El sistema presentará el certificado a obtener previo a su impresión.</p> <p>R.8.2 El sistema preguntará al estudiante si está de acuerdo con el valor que será acumulado en su cuenta por la emisión del certificado.</p> <p>R.8.4 El sistema notificará al estudiante por correo electrónico que se ha generado la deuda en su cuenta.</p> <p>R.8.1 El sistema generará la deuda correspondiente al certificado, en la cuenta del estudiante.</p> <p>R.11.1 El sistema imprimirá el certificado seleccionado por el usuario.</p> <p>R.11.3 El sistema colocará la firma digitalizada del/de la secretario(a) de la facultad a la que corresponde el estudiante.</p> <p>R.4.1 El sistema mostrará en el certificado los nombres y</p>

	<p>apellidos del estudiante.</p> <p>R.4.2 El sistema indicará en el certificado el número de cédula del estudiante.</p> <p>R.4.3 El sistema mostrará en el certificado la carrera y facultad del estudiante.</p> <p>R.15.1 El certificado del horario tendrá el formato del Anexo 5</p>
Requerimientos Especiales	

Caso de uso 16	Solicita Certificado del Pensum de la Carrera
Actores	Estudiante
Descripción	Se emitirá el certificado con la malla curricular del pensum y carrera que el estudiante este cursando.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El estudiante selecciona el Certificado del Pensum de la Carrera. 2. El sistema consulta los datos del pensum del estudiante para ser presentados en el certificado, mediante web service. 3. El sistema muestra una vista previa del certificado y el valor que se cargará a la cuenta del estudiante por la emisión. 4. El estudiante acepta los cargos por la emisión del certificado. 5. El sistema genera los cargos en la cuenta del estudiante. 6. El sistema Imprime el certificado. 7. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Flujo de Eventos Alternativo	<p>2a. Se produce un error al consultar los datos del pensum</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>4a. El estudiante no acepta los cargos por la emisión del certificado.</p> <ol style="list-style-type: none"> 1. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>5a. Se presenta un error al generar los cargos.</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Pre-condición	El estudiante debe identificarse
Post-condición	El sistema regresará al estado de espera
Caso de éxito	Se emitirá el certificado con el pensum del estudiante.
Triggers	Al seleccionar el certificado con el pensum de la carrera.
Requerimientos	R.1.1 El sistema verificará la identidad del usuario.

Funcionales	<p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p> <p>R.5.1 El sistema emitirá un certificado con el pensum académico del estudiante.</p> <p>R.10.1 El sistema presentará el certificado a obtener previo a su impresión.</p> <p>R.8.2 El sistema preguntará al estudiante si está de acuerdo con el valor que será acumulado en su cuenta por la emisión del certificado.</p> <p>R.8.1 El sistema generará la deuda correspondiente al certificado, en la cuenta del estudiante.</p> <p>R.8.4 El sistema notificará al estudiante por correo electrónico que se ha generado la deuda en su cuenta.</p> <p>R.11.1 El sistema imprimirá el certificado seleccionado por el usuario.</p> <p>R.11.3 El sistema colocará la firma digitalizada del/de la secretario(a) de la facultad a la que corresponde el estudiante.</p> <p>R.4.1 El sistema mostrará en el certificado los nombres y apellidos del estudiante.</p> <p>R.4.2 El sistema indicará en el certificado el número de cédula del estudiante.</p> <p>R.4.3 El sistema mostrará en el certificado la carrera y facultad del estudiante.</p> <p>R.5.2 El sistema ordenará el certificado de pensum por ciclos</p> <p>R.5.3 El sistema mostrará el código, nombre, número de créditos y prerrequisitos de cada materia en el certificado de pensum.</p> <p>R.16.1 El certificado del pensum académico tendrá el formato del Anexo 4</p>
Requerimientos Especiales	

Caso de uso 17	Solicita Certificado de notas históricas
Actores	Estudiante
Descripción	Se emitirá el certificado con las notas de estudiante organizadas de acuerdo al pensum académico.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El estudiante selecciona Certificado de notas Históricas 2. El sistema consulta los datos del pensum y las notas del estudiante para ser presentados en el certificado, mediante web service.

	<ol style="list-style-type: none"> 3. El sistema muestra una vista previa del certificado y el valor que se cargará a la cuenta del estudiante por la emisión. 4. El estudiante acepta los cargos por la emisión del certificado. 5. El sistema genera los cargos en la cuenta del estudiante. 6. El sistema Imprime el certificado. 7. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Flujo de Eventos Alternativo	<p>2a. Se produce un error al consultar los datos del estudiante</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>4a. El estudiante no acepta los cargos por la emisión del certificado.</p> <ol style="list-style-type: none"> 1. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>5a. Se presenta un error al generar los cargos.</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Pre-condición	El estudiante debe identificarse
Post-condición	El sistema regresará al estado de espera
Caso de éxito	Se emitirá el certificado de notas históricas.
Triggers	Al seleccionar el certificado de notas históricas.
Requerimientos Funcionales	<p>R.1.1 El sistema verificará la identidad del usuario.</p> <p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p> <p>R.6.1 El sistema emitirá un certificado con las notas históricas del estudiante.</p> <p>R.6.2 El sistema ordenará el certificado de notas históricas por ciclos.</p> <p>R.6.3 El sistema indicará si un estudiante aprobó una materia en primera, segunda o tercera matrícula en el certificado de notas históricas.</p> <p>R.6.4 El sistema indicará si la materia ha sido aprobada o reprobada en el certificado de notas.</p> <p>R.10.1 El sistema presentará el certificado a obtener previo a su impresión.</p> <p>R.8.2 El sistema preguntará al estudiante si está de acuerdo con el valor que será acumulado en su cuenta por la emisión del certificado.</p>

	<p>R.8.1 El sistema generará la deuda correspondiente al certificado, en la cuenta del estudiante.</p> <p>R.8.4 El sistema notificará al estudiante por correo electrónico que se ha generado la deuda en su cuenta.</p> <p>R.11.1 El sistema imprimirá el certificado seleccionado por el usuario.</p> <p>R.11.3 El sistema colocará la firma digitalizada del/de la secretario(a) de la facultad a la que corresponde el estudiante.</p> <p>R.4.1 El sistema mostrará en el certificado los nombres y apellidos del estudiante.</p> <p>R.4.2 El sistema indicará en el certificado el número de cédula del estudiante.</p> <p>R.4.3 El sistema mostrará en el certificado la carrera y facultad del estudiante.</p> <p>R.17.1 El certificado de notas históricas deberá contar con el promedio general de las materias indicadas</p> <p>R.17.2 El certificado de notas históricas mostrará los créditos extras tomados por el estudiante.</p> <p>R.17.3 El certificado de notas históricas indicará si cada materia fue homologada, convalidada o aprobada regularmente.</p> <p>R.17.4 El certificado de notas históricas tendrá el formato del Anexo 3</p>
Requerimientos Especiales	

Caso de uso 18	Solicita Certificado de notas actuales
Actores	Estudiante
Descripción	Se emitirá el certificado con las notas de las materias que esté tomando el estudiante en el periodo actual.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El estudiante selecciona Certificado de notas del actuales 2. El sistema consulta los datos del pensum y las notas del estudiante para ser presentados en el certificado, mediante web service. 3. El sistema muestra una vista previa del certificado y el valor que se cargará a la cuenta del estudiante por la emisión. 4. El estudiante acepta los cargos por la emisión del certificado. 5. El sistema genera los cargos en la cuenta del estudiante. 6. El sistema Imprime el certificado. 7. El sistema cierra la sesión del estudiante y regresa al estado de espera.

Flujo de Eventos Alternativo	<p>2a. Se produce un error al consultar los datos del estudiante</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>4a. El estudiante no acepta los cargos por la emisión del certificado.</p> <ol style="list-style-type: none"> 1. El sistema cierra la sesión del estudiante y regresa al estado de espera. <p>5a. Se presenta un error al generar los cargos.</p> <ol style="list-style-type: none"> 1. El sistema presenta un error. 2. El sistema cierra la sesión del estudiante y regresa al estado de espera.
Pre-condición	El estudiante debe identificarse
Post-condición	El sistema regresará al estado de espera
Caso de éxito	Se emitirá el certificado de notas actuales.
Triggers	Al seleccionar el certificado de notas actuales.
Requerimientos Funcionales	<p>R.1.1 El sistema verificará la identidad del usuario.</p> <p>R.2.2 El sistema emitirá certificados únicamente a estudiantes que se encuentren matriculados en la Universidad o a egresados.</p> <p>R.2.3 El sistema notificará al estudiante la razón por la que no puede solicitar un certificado.</p> <p>R.6.7 El sistema emitirá un certificado de las notas del estudiante en el periodo actual.</p> <p>R.6.4 El sistema indicará si la materia ha sido aprobada o reprobada.</p> <p>R.6.6 El sistema indicará en el certificado de notas actuales, las notas del periodo actual, correspondientes al aporte, examen, suspenso y nota final.</p> <p>R.10.1 El sistema presentará el certificado a obtener previo a su impresión.</p> <p>R.8.2 El sistema preguntará al estudiante si está de acuerdo con el valor que será acumulado en su cuenta por la emisión del certificado.</p> <p>R.8.1 El sistema generará la deuda correspondiente al certificado, en la cuenta del estudiante.</p> <p>R.8.4 El sistema notificará al estudiante por correo electrónico que se ha generado la deuda en su cuenta.</p> <p>R.11.1 El sistema imprimirá el certificado seleccionado por el usuario.</p> <p>R.11.3 El sistema colocará la firma digitalizada del/de la secretario(a) de la facultad a la que corresponde el estudiante.</p> <p>R.4.1 El sistema mostrará en el certificado los nombres y apellidos del estudiante.</p> <p>R.4.2 El sistema indicará en el certificado el número de</p>

	<p>cédula del estudiante.</p> <p>R.4.3 El sistema mostrará en el certificado la carrera y facultad del estudiante.</p> <p>R.18.1 El certificado de notas actuales mostrará el promedio de las calificaciones del periodo actual.</p> <p>R.18.2 El certificado de notas actuales tendrá el formato del Anexo 6</p>
<p>Requerimientos Especiales</p>	

5.3.2 Requerimientos No funcionales

Los requerimientos no funcionales están especificados de acuerdo a las seis características del modelo de calidad del software definidos en el estándar ISO/IEC 9126.

5.3.2.1 Funcionalidad

- Idoneidad:
 - o Se espera que todos los requerimientos funcionales estén operativos en un 100%.
- Precisión:
 - o Los certificados emitidos por el sistema deben reflejar la realidad actual del estudiante que los solicita.
- Interoperabilidad:
 - o El sistema debe interactuar fluidamente con el sistema de créditos de la Universidad, así como con su portal web. Sin producir errores ni interferencias entre dichos sistemas.
- Cumplimiento:
 - o El sistema contará con una interfaz de usuario que facilite su utilización mediante una pantalla táctil, y será diseñada de acuerdo a los estándares de interfaces de acceso estudiantil.
- Seguridad:
 - o El sistema verificará la identidad del usuario mediante su usuario y contraseña, antes de permitirle solicitar un certificado.
 - o El sistema obtendrá la información para la emisión de certificados mediante servicios WEB, lo que permitirá prevenir la modificación no autorizada a la base de datos de la Universidad.

5.3.2.2 Confiabilidad

- Madurez:
 - El sistema deberá mantener el margen de errores al mínimo.
 - Los errores del sistema no podrán afectar a otros sistemas de la Universidad.
- Tolerancia a Fallos:
 - El sistema deberá ser tolerante al ingreso erróneo del usuario y contraseña, sin bloquear la cuenta del estudiante.
- Recuperabilidad:
 - El sistema deberá recuperar de manera inmediata los datos afectados en caso de que se produzca un error en la solicitud del certificado.

5.3.2.3 Usabilidad

- Comprensibilidad:
 - El sistema deberá presentar un flujo de actividades para la obtención de certificados, de fácil comprensión para los estudiantes.
- Facilidad del Aprendizaje:
 - El sistema deberá ser altamente intuitivo, de tal manera que un estudiante que nunca ha utilizado el sistema pueda obtener rápidamente su certificado.
- Operatividad:
 - El sistema deberá contar con controles suficientemente grandes, que permitan la correcta operación de estos en la pantalla táctil.

- Atractivo:
 - o El sistema presentará una interfaz acorde a los colores de la Universidad del Azuay.
 - o El sistema mostrará una pantalla de inicio o bienvenida en su estado de espera, la cual capte la atención de los estudiantes e indique su cuál es su uso.

5.3.2.4 Eficiencia

- Tiempo:
 - o La vista previa del certificado deberá ser presentada en un máximo de 3 segundos.
- Uso de Recursos:
 - o El sistema se ejecutará en un Kiosco táctil con impresor incluido con un procesador de mínimo 1.6 GHZ.
 - o El sistema hará uso de los servicios Web que accederán a una base de datos Oracle.
 - o Se diseñará el sistema para que un estudiante obtenga su certificado en un máximo de 4 minutos.

5.3.2.5 Mantenibilidad

- Analizabilidad:
 - o El sistema indicará claramente al usuario el tipo de error presentado.
- Variabilidad:
 - o El sistema deberá contar con la documentación requerida para facilitar los cambios futuros y la inclusión de nuevos certificados.

- El sistema manejará los procesos comunes a todos los certificados de manera separada, para facilitar la implementación de nuevos tipos de certificados en el sistema.
 - El sistema contará con un componente que permita el manejo de distintos tipos de certificados, adaptándose a los nuevos que sean creados.
 - El sistema implementará web services que permitan la utilización de los datos desde diferentes sistemas y la creación de nuevos certificados.
 - El sistema tendrá separada la parte operativa de la parte gráfica para facilitar la implementación de otro tipo de interfaz.
- Estabilidad:
 - El sistema deberá ser exhaustivamente probado después de cada modificación, para evitar efectos inesperados.
 - Capacidad de Prueba:
 - Se deberá contar con un plan de pruebas completo que permita asegurar que el sistema o sus modificaciones cumplen los requerimientos planteados en todos los casos.

5.3.2.6 Portabilidad

- Adaptabilidad:
 - El sistema será desarrollado en JAVA, lo que permitirá su ejecución en cualquier plataforma que ejecute la máquina virtual de JAVA.
 - El sistema tendrá separada la parte operativa de la parte gráfica para facilitar la implementación de otro tipo de interfaz.

- Facilidad de Instalación:
 - o El sistema constará de un archivo que servirá de ejecutable genérico para cualquier plataforma. Para su ejecución el kiosco deberá tener instalado el Java JRE más reciente.
- Conformidad:
 - o El sistema hará uso de las funcionalidades de la plataforma Java que funcionan en todos los sistemas operativos.
 - o El look and feel utilizado en la interfaz gráfica del sistema deberá ser Nimbus, el cual no corresponde a ningún Sistema Operativo en específico.
- Intercambiabilidad:
 - o El sistema permitirá ser adaptado para ser usado en la secretaría de la facultad para generar reportes de estos certificados.
 - o El sistema tendrá separada la parte operativa de la parte gráfica para facilitar la implementación de otro tipo de interfaz.

5.4 Interfaces Externas

5.4.1 Interfaces del Software

El sistema debe comunicarse con: el sistema de créditos de la Universidad, en lo correspondiente a matriculas, calificaciones, carreras, etc. Al sistema de tesorería, para la generación de deudas, y al sistema de usuarios, para comprobar el usuario y contraseña. Para dicho efecto se utilizarán servicios web, que permitan obtener los datos sin comprometer la seguridad de la base de datos.

5.4.2 Interfaces de usuario

El sistema deberá contar con una interfaz de usuario que facilite el uso del sistema mediante su pantalla táctil, siendo lo suficientemente intuitiva para que cualquier usuario la pueda manejar en su primera experiencia con el sistema. Los controles gráficos deberán ser suficientemente grandes para facilitar el uso mediante el tacto.

La interfaz general del sistema deberá ser diseñada con los colores y logos de la Universidad del Azuay, manteniendo el estándar de la interfaces gráficas de acceso estudiantil.

5.4.3 Interfaces de Comunicaciones

La comunicación del kiosco con los sistemas de la Universidad del Azuay se la realizará mediante servicios web.

En este capítulo se plasma el análisis del sistema propuesto, en base a una especificación de requisitos extendida. Los requisitos funcionales fueron agrupados por casos de uso. En donde, cada caso de uso cuenta con una especificación detallada y el flujo de actividades correspondiente. Por su parte, los requerimientos no funcionales fueron descritos en función del estándar ISO/IEC 9126. Además, se realizó una breve descripción de las interfaces externas del sistema. Este análisis servirá de base para el diseño del sistema propuesto, el cual se detalla en el siguiente capítulo.

CAPÍTULO VI

6. DISEÑO DEL SISTEMA PROPUESTO

En base a los conceptos sobre kioscos, interfaz gráfica táctil y servicios web, presentados en los capítulos II, III y IV; y, en base al análisis realizado en la Especificación de Requisitos del Software del capítulo V, se presenta el Diseño del Sistema para kioscos emisores de Certificados Académicos. En este capítulo se detalla: el Diseño Orientado a Objetos DOO, plasmado mediante diagramas UML; y, el diseño de la Interfaz gráfica (GUI) del kiosco, basada en los lineamientos presentados en el capítulo III.

6.1 Diseño del sistema mediante UML

UML es un lenguaje de modelado estándar multipropósito que permite especificar, visualizar, construir y documentar artefactos en sistemas software. UML fue creado para solventar la inexistencia de un lenguaje de modelado estándar para los conceptos de orientación a objetos, bajo el argumento de que “una imagen vale más que mil palabras”. Su objetivo es el de ser un lenguaje fácil de entender y utilizar para todos los modeladores, e incluso para los usuarios del negocio, gente común o cualquiera que esté interesado en el funcionamiento del sistema.

La razón para realizar el modelado del sistema propuesto es que “una organización exitosa es aquella que constantemente implementa software de calidad, que cumple con los requerimientos de sus usuarios”³⁵. Teniendo en cuenta que la única manera de obtener software de calidad es mediante un modelado adecuado.

³⁵ (Boochm, Rumbaugh y Jacobson)

6.1.1 Conceptos de Orientación a Objetos

“UML puede ser descrito como el sucesor del Análisis y Diseño Orientado a Objetos”³⁶.

Un objeto contiene datos, que representan el estado del objeto, y métodos, que permiten controlar estos datos. Así, una clase es una abstracción de un objeto. Las clases forman jerarquías que son representadas mediante la herencia.

UML es un lenguaje poderoso que permite representar todos los conceptos de Análisis y Diseño Orientado a Objetos. Los objetos, que con entidades del mundo real, y los conceptos, tales como la abstracción, encapsulación, herencia, polimorfismo; pueden ser representados utilizando UML.

Algunos de los conceptos fundamentales de la orientación a objetos son detallados brevemente a continuación:

- **Objetos:** representan una entidad y son la unidad básica de un bloque.
- **Clase:** es la abstracción o estructura genérica de un tipo de objeto.
- **Abstracción:** representa el comportamiento de una entidad del mundo real.
- **Encapsulación:** mecanismo que empaqueta los datos de una manera conjunta y oculta del resto del mundo.
- **Herencia:** mecanismo que permite crear nuevas clases de una existente, heredando sus atributos y comportamiento.
- **Polimorfismo:** define el mecanismo para la existencia en diferentes formas.

6.1.2 Análisis y Diseño Orientado a Objetos

El AOO y DOO puede ser resumido en 3 fases:

³⁶ (TutorialsPoint)

- En el AOO primero se identifican los objetos y sus relaciones, para después, describirlos de una manera apropiada. Los objetos deben ser identificados junto con sus responsabilidades, que son las acciones a ser tomadas por el objeto. Cuando estas responsabilidades se llevan de una manera colaborada, se alcanza el propósito del sistema.
- La segunda fase es el DOO, en él, se hace énfasis en los requerimientos y en su cumplimiento. Los objetos deben colaborar de acuerdo a su asociación.
- Como última fase se tiene la implementación de los diseños en un lenguaje orientado a objetos.

6.1.3 Arquitectura UML

Todo sistema es utilizado por diferentes usuarios. Estos usuarios pueden ser desarrolladores, personal de pruebas, gente de negocios, analistas, expertos, usuarios comunes, etc. Por esta razón, la arquitectura del sistema propuesto fue realizada tomando en cuenta diferentes perspectivas. Mientras más entendemos el sistema, mejor será realizado.

UML toma un rol importante en definir las perspectivas de un sistema. Estas perspectivas UML son las utilizadas en el sistema emisor de certificados académicos y son:

- **Diseño:** el diseño del sistema consiste en clases, interfaces y colaboración. Para ello, UML tiene los diagramas de clases y de objetos.
- **Implementación:** define los componentes que se ensamblan juntos para completar el sistema físico. Para ello se utiliza el diagrama de componentes.
- **Proceso:** define el flujo del sistema, por lo cual, se utilizan los mismos diagramas de clases y de objetos.

- Despliegue: representa los nodos físicos del sistema que forman el hardware. Esta perspectiva está representada en el diagrama de despliegue.

En el centro de estas perspectivas se encuentra la vista de casos de uso, que contiene las 4 anteriores. El caso de uso representa la funcionalidad de un sistema. El diagrama de casos de uso del sistema propuesto es el que se muestra en la Figura 5-2.

6.2 Modelado UML

Existen diferentes diagramas para cada tipo de modelado UML. Los tres tipos de modelados más importantes son:

6.2.1 Modelado Estructural

Los diagramas que lo conforman capturan las características estáticas del sistema y representan el framework del sistema. Consiste en los siguientes:

- Diagrama de Clases.
- Diagrama de Objetos.
- Diagrama de Despliegue.
- Diagrama de Paquetes.
- Diagrama de Estructura compuesta.
- Diagrama de Componentes.

Para el modelado estructural del sistema propuesto se realizaron los diagramas de clases, de despliegue y de paquetes, los cuales serán descritos en las siguientes secciones.

6.2.1.1 Diagrama de Clases

El diagrama de clases representa la vista estática de una aplicación, y no es solamente utilizado para visualizar, describir y documentar diferentes aspectos del sistema, también, para construir código ejecutable de la aplicación. El diagrama de clases describe los atributos y operaciones de una clase, así como las restricciones del sistema.

El propósito del diagrama de clases dentro de la aplicación propuesta fue:

- Analizar y diseñar una vista estática de la aplicación.
- Describir responsabilidades o funciones del sistema.
- Ser una base para los diagramas de despliegue y de paquetes.
- Ser la base para el desarrollo del código.

El primer diagrama de clases desarrollado es el de la Figura 6-1, y correspondiente a la parte de los web Services o servicios web. En él, se pretende representar los servicios o métodos que se prestarán por medio del servicio web, y las clases que permiten empaquetar las respuestas de estos servicios.

Como se observa en la Figura 6-1, las clases ubicadas en la parte inferior son las que permiten realizar el empaquetado de la respuesta a cada uno de los servicios y funcionan como estructuras de datos o registros. Los métodos consultarDatosEstudiante y consultarDatosEgresado de la clase WebServices, son los únicos que utilizan estas clases para devolver un único registro. El resto de métodos, en cambio, utilizan estas clases para devolver varios registros. Por ejemplo, el método consultarNotasHistoricas da como resultado un conjunto de las materias tomadas, representadas por los campos de la clase MateriasTomadas.

Mientras la clase WebServices se ocupa de todos los servicios relacionados a la obtención de datos para el certificado, la clase TesoreriaWebServices posee los

servicios para la consulta del costo del certificado y para acumular su valor en la cuenta del estudiante. Cabe recalcar que esta clase deja abierta la posibilidad de crear distintos costos a futuro para los distintos certificados. Sin embargo, esto no será utilizado en la aplicación propuesta, pues todos los certificados tendrán el mismo costo.

En cuanto a la utilización de estos servicios web para la creación de certificados, se elaboró el diagrama de clases de la Figura 6-2. La clase Estudiante hereda los atributos de la clase Estudiante de la Figura 6-1, e incorpora métodos para utilizar los servicios de consulta de datos del estudiante, verificar si el estudiante está egresado o matriculado y verificar la identidad del usuario. El método verificarIdentidad utiliza el método privado consultarDatosEstudiante para, luego de verificar su identidad por el servicio web existente del sistema de usuarios, entregar la información completa del estudiante con el usuario solicitado.

La clase Certificado (Figura 6-2), por su parte, utiliza los métodos de la clase estudiante para discernir, mediante el método obtenerCertificadosDisponibles, qué certificados estarán disponibles, dependiendo si el estudiante está matriculado o es egresado. Los certificados disponibles serán retornados como objetos de la clase Certificado, es decir, junto con el nombre que tenga el certificado, la dirección del archivo que contenga su estructura y una imagen para representar el certificado.

La clase certificado, además, incorpora los métodos generarCertificadoPDF y generarCertificadoHTML, que solicitan la creación del certificado solicitado en el formato deseado, devolviendo el nombre con el que fueron generados. En un diseño inicial se tenía únicamente la generación del certificado en general, pero por razones de implementación del diseño, se requirió separarlos en dos métodos.

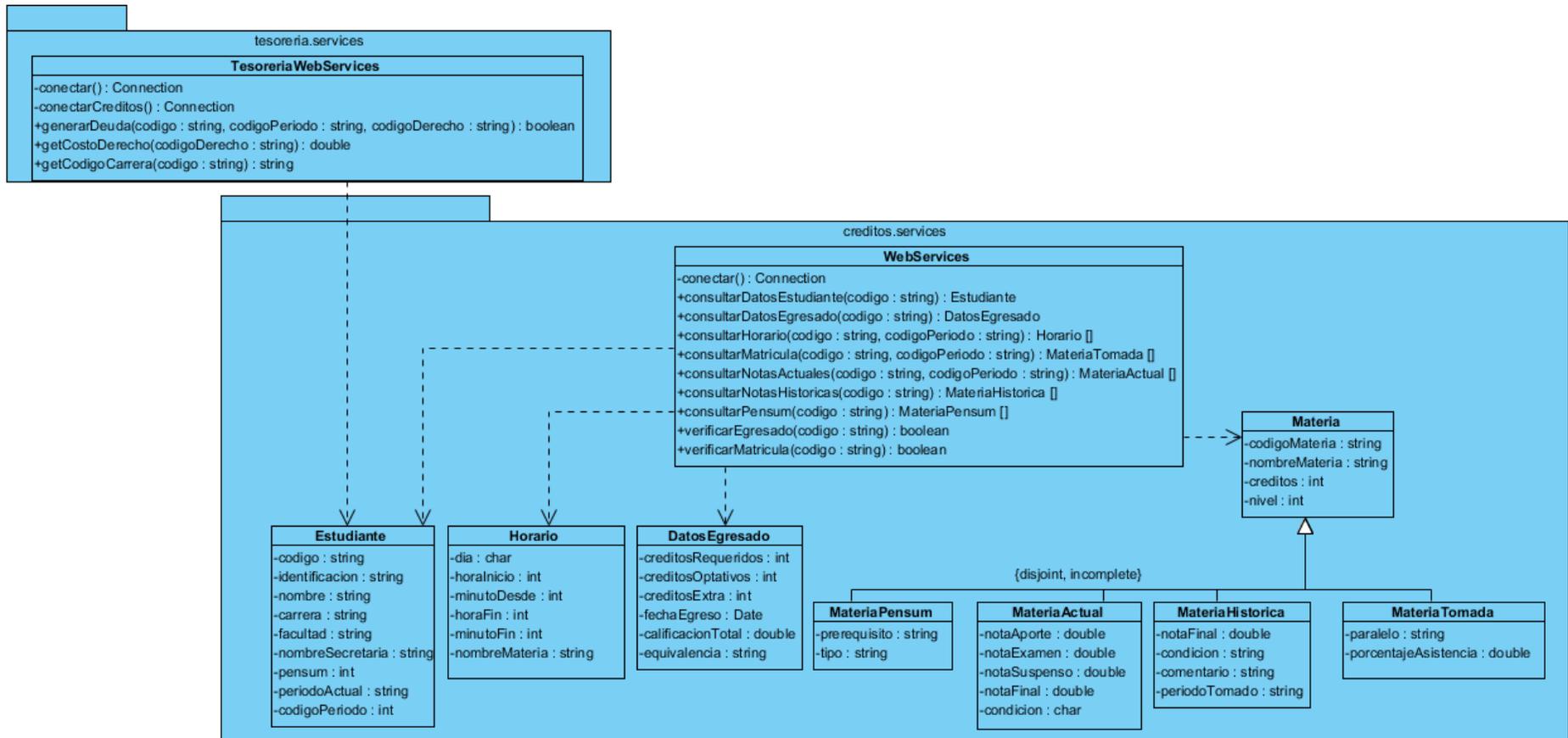


Figura 6-1: Diagrama de Clases de los web services.

En estos métodos, se optó colocar como parámetro una clase que permita la inclusión de varios parámetros a gusto de la interfaz gráfica, pues, de solicitar como parámetros únicamente el código del estudiante y del periodo actual (que son los únicos necesarios para los certificados implementados), quedaría restringida la creación de nuevos certificados a aquellos que requieran únicamente estos dos parámetros.

El método `iniciarPlataforma` permite que la plataforma que genera los certificados sea iniciada con anterioridad, ayudando a que los certificados se generen con celeridad.

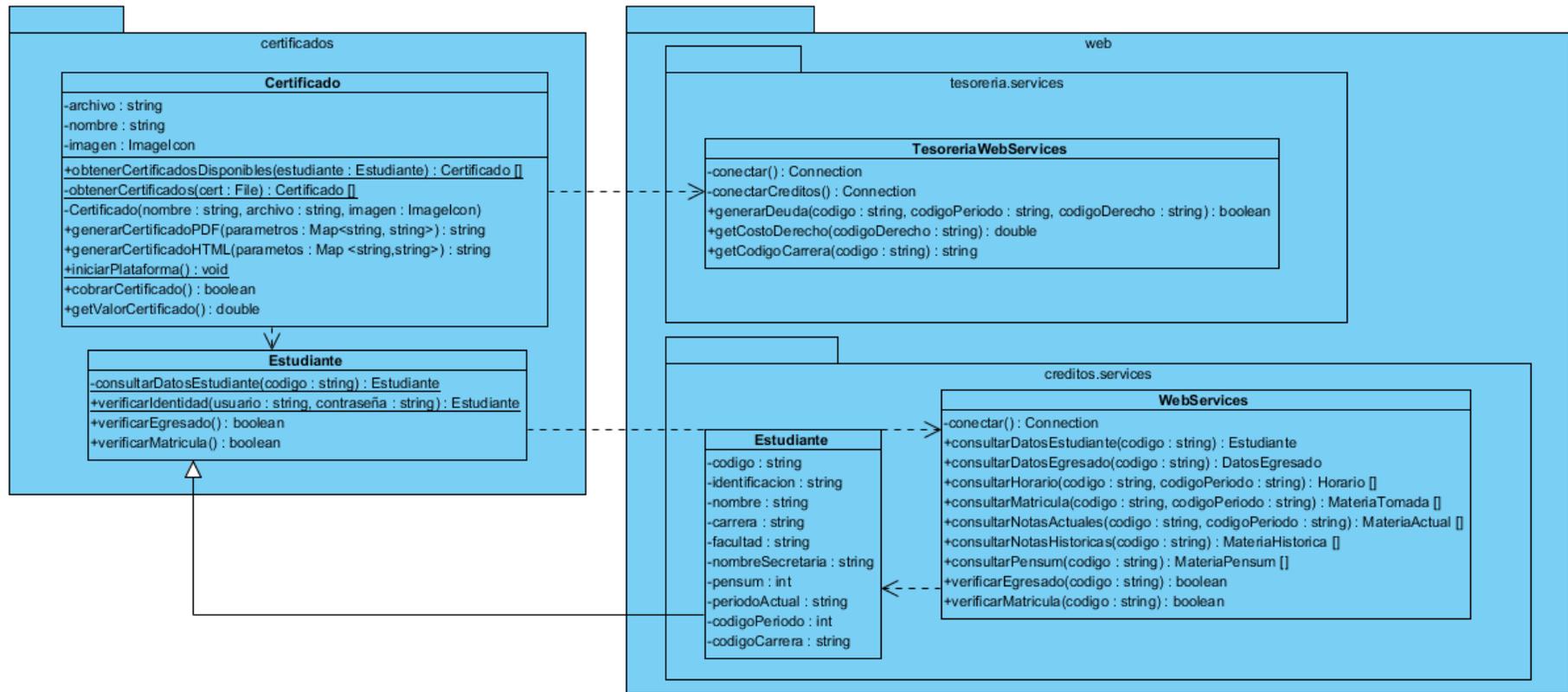


Figura 6-2: Diagrama de Clases de la generación de certificado

6.2.1.2 Diagrama de Paquetes

El diagrama de paquetes es un diagrama estructural que muestra los paquetes y las dependencias entre ellos. Estos diagramas permiten mostrar diferentes puntos de vista del sistema. (UML diagrams)

Los paquetes muchas veces se utilizan para organizar clases, en cuyos casos se siguen las siguientes reglas (Ambler):

- Generalmente, clases de la misma jerarquía de herencia van en el mismo paquete.
- Clases relacionadas mediante composición a menudo pertenecen al mismo paquete.
- Clases que colaborar mucho entre sí, generalmente pertenecen al mismo paquete.

Una pequeña muestra de los paquetes del sistema propuesto ha sido mostrada en los diagramas de la Figura 6-1 y Figura 6-2. Tomando en cuenta las clases que se encuentran en cada paquete, se presenta el diagrama de paquetes de la Figura 6-3.

Los paquetes *web* y *certificados* del paquete *uda* son los presentados en la Figura 6-1 y Figura 6-2 respectivamente. El paquete *certificados*, como se mostró en el diagrama de clases de la Figura 6-2, hace uso de los servicios web del paquete *web*. Además, este paquete hace uso de una librería externa representada por el paquete “*org.Eclipse.BIRT*”, que corresponde a las librerías del generador de reportes BIRT, necesario para generar los certificados en PDF y HTML.

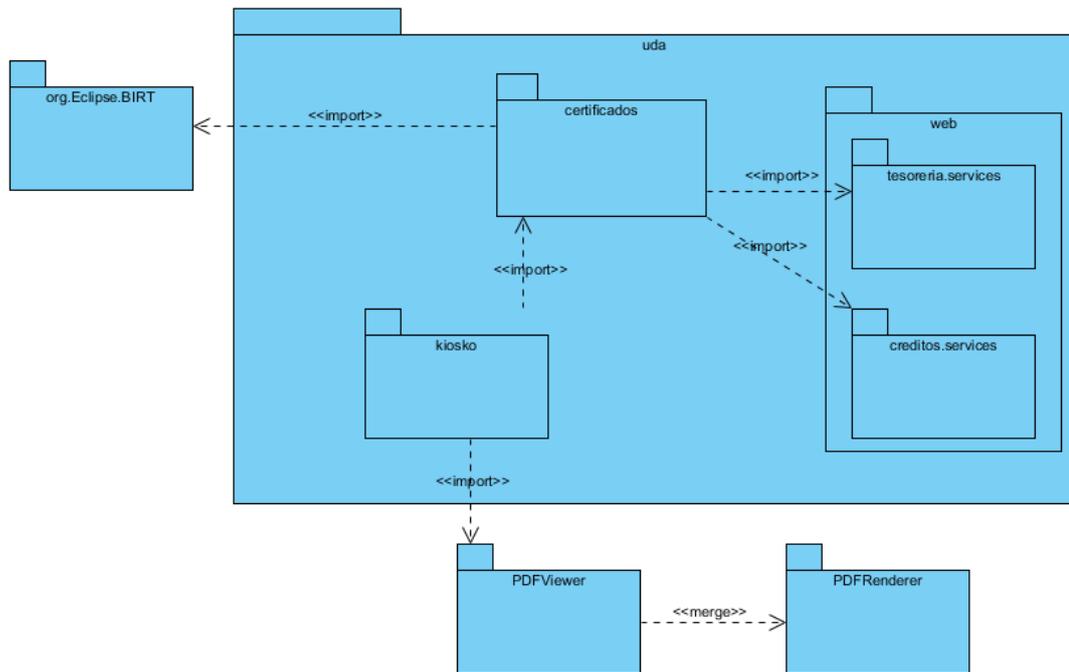


Figura 6-3: Diagrama de Paquetes.

El paquete *kiosko* representa la interfaz gráfica del kiosco y las clases que la compongan. Este paquete hace uso del paquete *PDFViewer*, el cual que le permite visualizar un documento PDF en la interfaz gráfica e imprimirlo. Para ello, el paquete *PDFViewer* hace uso de una librería *PDFRenderrer* que le provee esta funcionalidad.

El sistema emisor de certificados académicos fue diseñado separando los servicios web (paquete *web*) de la generación de certificados (paquete *certificados*), para que los servicios puedan ser utilizados en otros sistemas. La generación de certificados, a su vez, fue diseñada separada de la interfaz gráfica (paquete *kiosko*) para que en un futuro se pueda llevar el sistema de certificados a la web (mediante el método *generarCertificadoHTML* de la clase *certificado*) y sólo se requiera modificar la interfaz gráfica, es decir, se crearía un nuevo paquete que contenga la interfaz gráfica y que haga uso del paquete *certificados*. Esta portabilidad está descrita a más detalle en el modelado arquitectónico explicado más adelante.

6.2.1.3 Diagrama de Despliegue

Los diagramas de despliegue se utilizan para visualizar la topología de los componentes físicos de un sistema, en donde los componentes de software son desplegados o ejecutados. El diagrama de despliegue en el sistema propuesto es utilizado para:

- Visualizar la topología del hardware del sistema.
- Describir qué componentes de hardware se utilizan para ejecutar los componentes de software.
- Describir los nodos de ejecución.

El diagrama de despliegue del sistema propuesto es el de la Figura 6-4. Este diagrama indica qué artefactos de software se utilizan, en qué equipos y bajo qué plataforma deben ejecutarse. Los artefactos ya fueron presentados mediante el diagrama de paquetes de la Figura 6-3.

En el primer nodo, se representa la base datos de la Universidad del Azuay, que se encuentra en el Gestor de Base de datos Oracle. La base de datos es accedida por TCP/IP mediante la librería OJDBC.jar, que es utilizada, a su vez, por los artefactos `creditos.war` y `tesorería.war` que corresponden a los paquetes `uda.web.creditos.services` y `uda.web.tesoreria.services`, respectivamente. Estos artefactos son ejecutados dentro de un servidor Glassfish por ser componentes web y se manifiestan como servicios web.

El nodo kiosco representa el equipo que se encargará de emitir los certificados. El artefacto `Certificados.jar` representa al paquete `uda.certificados` que hace uso de las librerías BIRT y toma los datos de los servicios web. El artefacto `kiosco.jar` representa a la interfaz gráfica del kiosco, que utiliza el artefacto `Certificados.jar` para generar los certificados y `PDFViewer.jar` para mostrarlos en pantalla e imprimirlos. Todos los artefactos de este nodo kiosco se ejecutan bajo el ambiente JRE de Java.

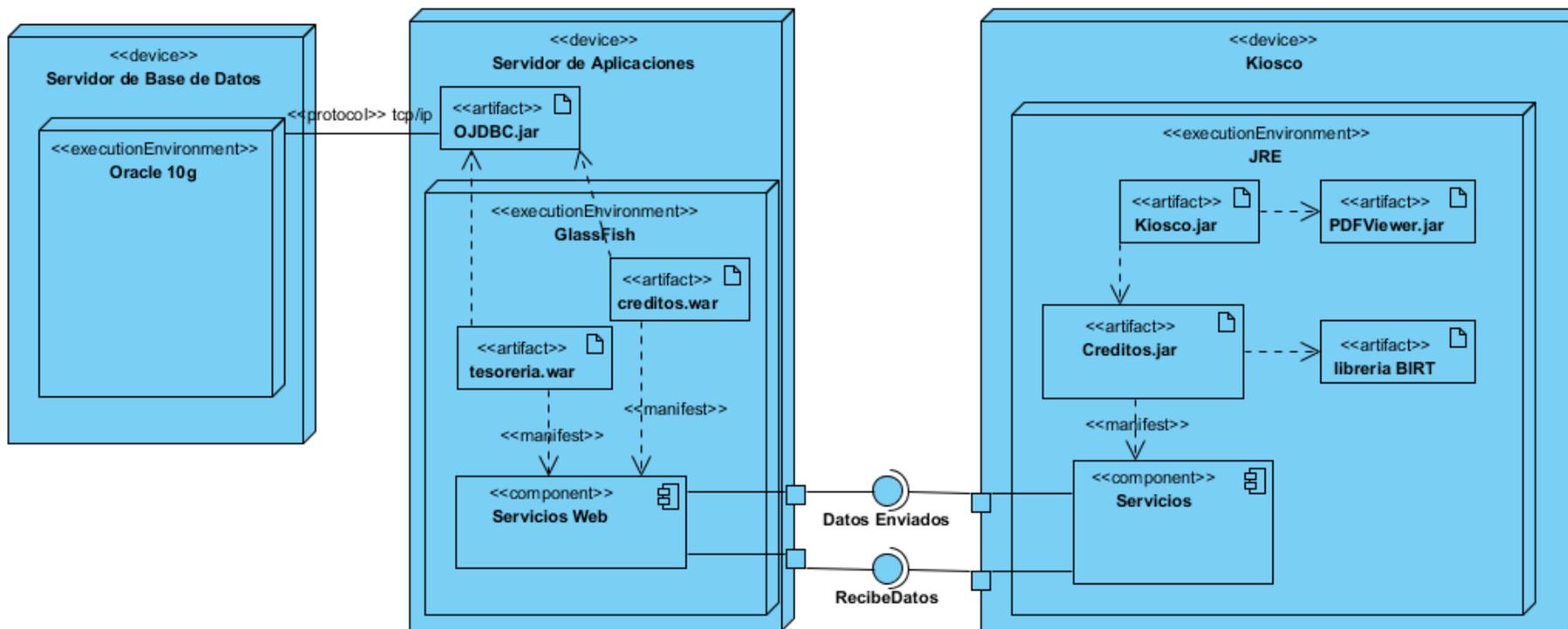


Figura 6-4: Diagrama de Despliegue

6.2.2 Modelado de Comportamiento

Al modelar un sistema, el aspecto más importante es capturar el comportamiento dinámico, es decir, el comportamiento cuando el sistema está en funcionamiento. Los modelados de comportamiento describen la interacción existente en el sistema y representan la interacción entre los diagramas estructurales. Los diagramas de comportamiento son:

- Diagramas de Actividad.
- Diagramas de Estados.
- Diagramas de Interacción (de secuencia y de colaboración).
- Diagramas de Casos de Uso.

En el sistema propuesto se realizaron los diagramas de Actividad (Figura 5-3) y de Casos de Uso (Figura 5-1 y Figura 5-2), mostrados en el capítulo anterior. Además, para una mejor descripción del comportamiento, se efectuaron los diagramas de secuencia y de estados, los cuales serán descritos en las siguientes secciones.

6.2.2.1 Diagrama de Casos de Uso

El diagrama de casos de uso es uno de los 5 diagramas que modelan el comportamiento dinámico del sistema. Este diagrama consiste de actores, casos de uso y relaciones. Debido a que los otros cuatro diagramas tienen el mismo propósito, los diagramas de casos de uso se los utiliza específicamente para:

- Reunir requerimientos de un sistema incluyendo influencias internas y externas.
- Dar una vista externa del sistema.
- Identificar factores internos y externos que influyen en el sistema.
- Mostrar la interacción entre los requerimientos y los actores.

Tomando en cuenta estos propósitos, en especial el referente al análisis de requerimientos y diseño de alto nivel, se utilizó el diagrama de casos de uso para agrupar y organizar los requisitos en la Especificación de Requisitos de Software del Capítulo V. Primero, se desarrolló un diagrama de casos de uso simple e inicial, que represente de una manera general la funcionalidad del sistema, este diagrama es el que se muestra en la Figura 5-1. El caso de uso central del diagrama es la solicitud del certificado, que es el objetivo principal del sistema. Además, el diagrama muestra la interacción que debe tener la aplicación a desarrollar con el Sistema Informático general de la Universidad del Azuay.

Posteriormente, el diagrama de la Figura 5-1 se refina y extiende, formando el diagrama de casos de uso de la Figura 5-2. Este diagrama extendido descompone el sistema de la Universidad del Azuay en el sistema de créditos, el sistema de tesorería y el de usuarios. Además, separa la parte de recolección de información, que será llevada a cabo mediante los servicios web, los cuales, permitirán la comunicación entre el sistema propuesto y los sistemas de la Universidad. Adicionalmente, se especifican los tipos de certificados que serán implementados en primera instancia, y los servicios web estos que utilizarán.

6.2.2.2 Diagrama de Actividad

El diagrama de actividad es básicamente un diagrama de flujo que representa el flujo de una actividad hacia otra, en donde una actividad puede ser descrita como una operación del sistema. El flujo entre actividades puede ser secuencial, ramificado o concurrente.

Los otros 4 diagramas de comportamiento muestran el flujo de mensajes entre un objeto a otro, pero el diagrama de actividad se especializa en mostrar el flujo de mensajes de una actividad a otra. Este diagrama tiene como propósito:

- Mostrar el flujo de actividades de un sistema.
- Describir la secuencia de una actividad a otra.
- Describir el flujo secuencial, ramificado y concurrente de un sistema.

El diagrama de actividad del sistema propuesto es el presentado en la Figura 5-3, el diagrama fue utilizado en el capítulo anterior para describir el flujo general de actividades al utilizar el kiosco, desde el inicio de sesión hasta la obtención del certificado impreso. Esto fue necesario para describir el flujo de eventos detallado en la descripción de cada caso de uso.

El diagrama, además, permite conocer que actor o sistema es el responsable de llevar acabo cada actividad del proceso. Esto se logra mediante el uso de los carriles Usuario, Estudiante, Sistema de Certificados y Sistema de créditos, este último, representa a los sistemas de créditos, de usuarios y de tesorería.

6.2.2.3 Diagramas de Interacción – Los Diagramas de Secuencia

Los diagramas de interacción, como su nombre los dice, representan las interacciones entre diferentes elementos del modelo. Este comportamiento interactivo es representado en UML por dos diagramas, conocidos como el diagrama de secuencia y el diagrama de colaboración.

El propósito de estos diagramas es similar: capturar el comportamiento interactivo, pero desde distinto ángulo. El diagrama de secuencia enfatiza el tiempo en el que se da la secuencia de los mensajes, mientras el diagrama de colaboración, la manera en que los objetos y actores colaboran entre sí.

En el diseño del sistema propuesto se utilizó únicamente diagramas de secuencia, pues se consideró que muestran de mejor manera el comportamiento interactivo del sistema, y aclaran el orden en la que se realizan las interacciones. Por su parte,

el diagrama de colaboración no brindaba mayor información adicional, de la ya obtenida con los diagramas de secuencia, y el resto de diagramas de comportamiento.

Los diagramas de secuencia fueron construidos con el propósito de:

- Capturar el comportamiento interactivo del sistema.
- Describir el flujo de mensajes y su orden.
- Describir la interacción entre objetos.

Para el sistema emisor de certificados académicos, primero se realizó un diagrama de secuencia inicial, que contiene toda la secuencia de funcionamiento del sistema, desde que el usuario ingresa sus datos hasta que el estudiante recibe su certificado. Este diagrama es el presentado en la Figura 6-5.

Al realizar un diseño más detallado de la interacción dinámica entre los objetos, se necesitó separar las secuencias de:

- Inicio de sesión y visualización de certificados disponibles.
- Selección y visualización del certificado.
- Impresión y cobro del certificado.

El diagrama de secuencia que indica el proceso para llevar a cabo el inicio de sesión y la visualización de certificados disponibles es el de la Figura 6-6. En este diagrama, se muestran los actores de esta parte del proceso, las clases y objetos involucrados, los métodos llamados y las respuestas a estos métodos. El objeto Créditos de la clase WebServices es el servicio web, y este, extrae la información de la base de datos del Sistema de Créditos de la Universidad del Azuay.

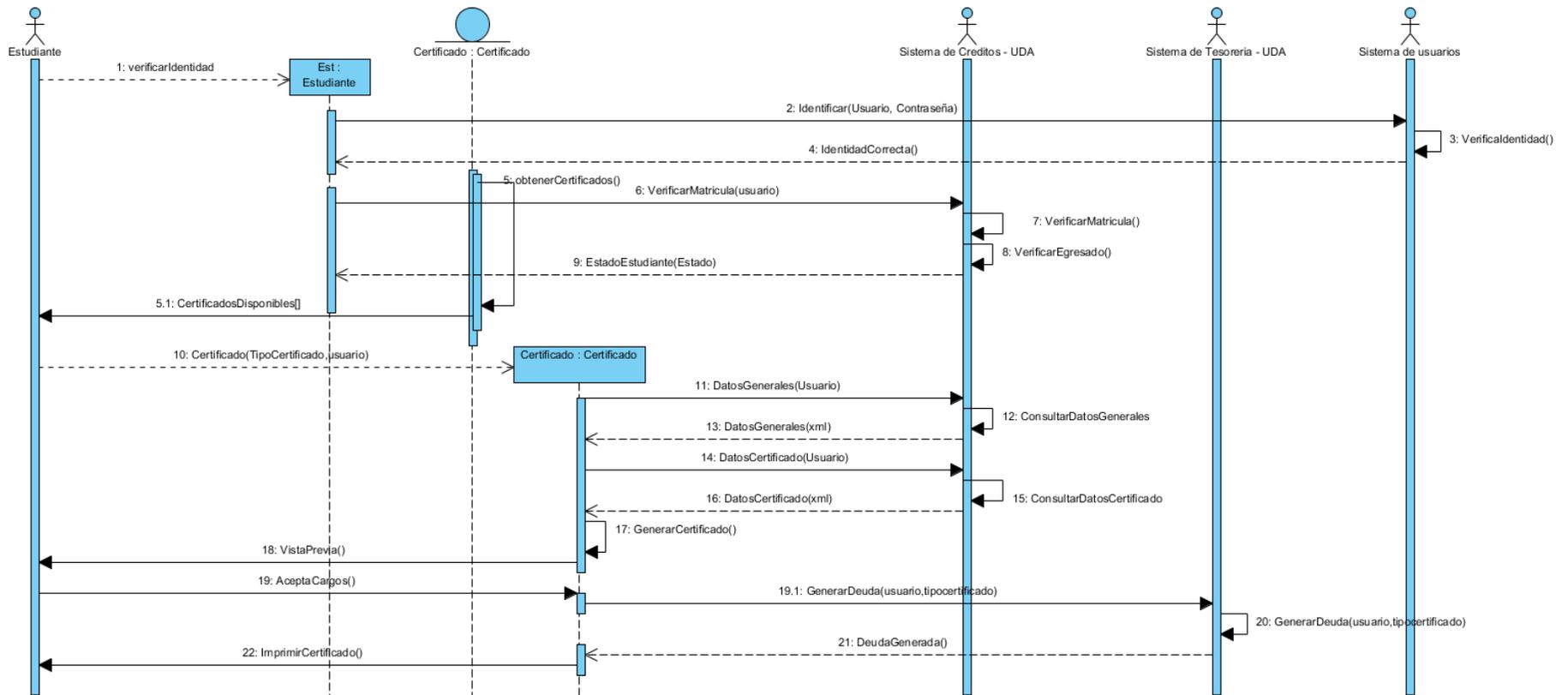


Figura 6-5: Diagrama de Secuencia General.

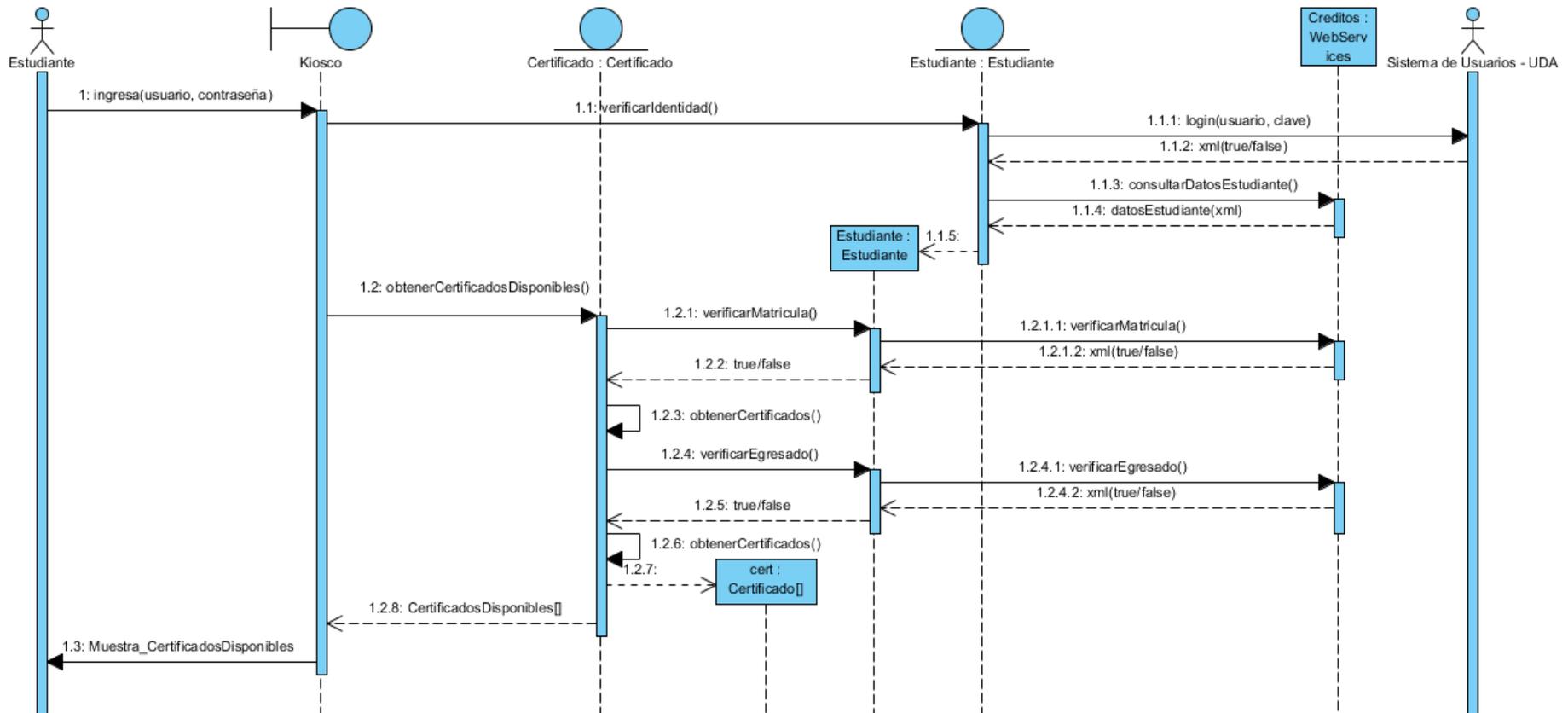


Figura 6-6: Diagrama de Secuencia para Identificar Estudiante e indicar los certificados disponibles.

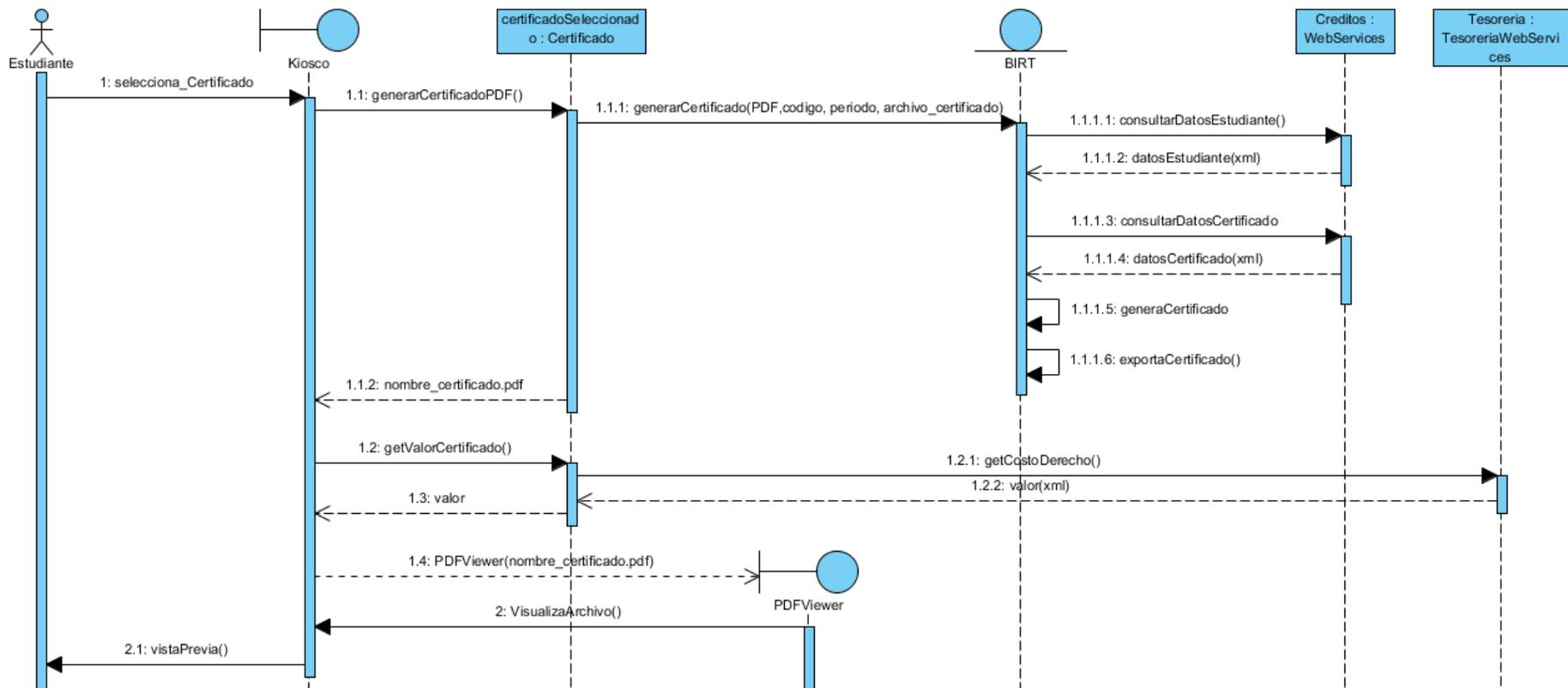


Figura 6-7: Diagrama de Secuencia para generar certificados y visualizarlos.

El diagrama de la Figura 6-7, describe la secuencia de métodos a llamar para generar el certificado solicitado por el estudiante y mostrarlo en pantalla. Este diagrama permite mostrar la dependencia que tiene el sistema de la plataforma de reportes BIRT. Esta plataforma, es llamada cuando se invocan los métodos `generarCertificadoPDF` o `generarCertificadoHTML`, como se mencionó anteriormente. En este diagrama, también se incluye la interacción con la Clase `TesoreriaWebService`, que realiza las consultas en el Sistema de Tesorería de la Universidad del Azuay, y con la interfaz `PDFViewer` que permite visualizar el certificado en PDF.

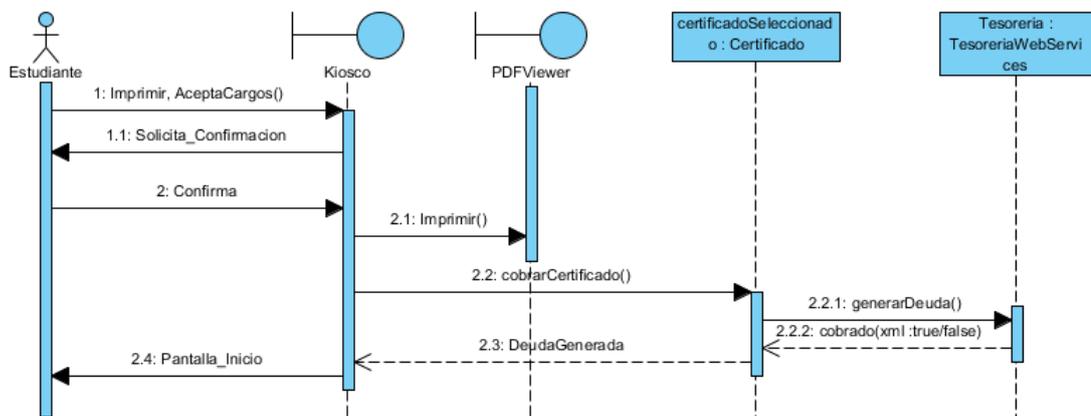


Figura 6-8: Diagrama de Secuencia de la impresión y generación de la deuda del certificado.

Finalmente, el comportamiento del sistema, de sus objetos y del servicio de web de tesorería, en lo correspondiente a la impresión y generación de la deuda del certificado en la cuenta del estudiante, se plasma en el diagrama de secuencia de la Figura 6-8. Como se observa, la interfaz `PDFViewer` es la que proporciona la impresión del certificado.

6.2.2.4 Diagrama de Estados

El diagrama de estados describe una máquina de estados, es decir, una máquina que define diferentes estados de un objeto y estos estados son controlados por

eventos internos y externos. El diagrama de estados describe el flujo de un estado hacia otro.

El diagrama de estados de la Figura 6-9, es el utilizado para describir los estados del sistema emisor de certificados que se ejecuta en un kiosco. Estos estados nos permiten visualizar de mejor manera el comportamiento del sistema, conocer los eventos que causan un cambio de estado, y son una base para el diseño de la interfaz gráfica del sistema en cada estado.

El estado “en espera” representa cuando la aplicación muestra la pantalla de bienvenida e indica que en el kiosco se pueden obtener los certificados. Cuando el usuario toca la pantalla del kiosco, se pasa al estado “esperando identificación”, en donde se muestra la pantalla para el ingreso de usuario y contraseña.

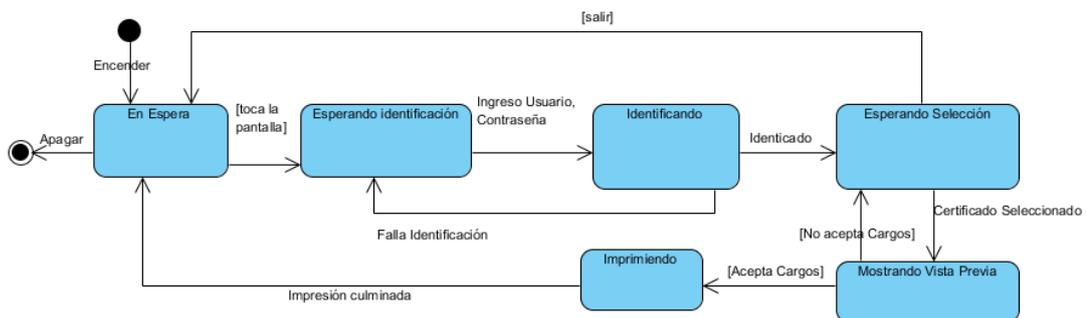


Figura 6-9: Diagrama de Estados del Kiosco emisor de certificados

Si el usuario ingresa el usuario y contraseña, el sistema pasa al estado de “identificado”, para verificar la identidad del usuario. Si el usuario y contraseña son correctos, el sistema pasa a “esperando selección”, donde se muestran los certificados que el estudiante puede solicitar. Al seleccionar un certificado, el sistema pasa a “mostrando vista previa”. Si el estudiante acepta los cargos, el flujo se dirige al estado de “imprimiendo”. Luego de finalizada la impresión, el sistema regresa al estado de “en espera”, mostrando nuevamente la pantalla inicial.

6.2.3 Modelado Arquitectónico

El modelado arquitectónico representa el marco general del sistema, contiene elementos estructurales y de comportamiento. Para su representación se utiliza el diagrama de paquetes.

Para el sistema propuesto, a más del diagrama de paquetes de la Figura 6-3, explicado en el apartado 6.2.1.2, se ha desarrollado el diseño arquitectónico de la Figura 6-10. Como se observa, el sistema emisor de certificados académicos tiene como base los servicios web, la idea de utilizar los servicios web es que estos puedan ser usados para la generación de certificados y para otros propósitos o aplicaciones futuras como apps para móviles.

El componente “certificados” es el encargado de conectarse con los servicios web, generar los certificados y proveer la funcionalidad a la aplicación. El diseño arquitectónico de este componente permite que su funcionalidad sea utilizada por diferentes interfaces gráficas. Una de ellas, es la interfaz gráfica del kiosco interactivo que es el propósito de esta tesis.



Figura 6-10: Diseño arquitectónico.

A su vez, el componente “Certificados”, como se indica en la Figura 6-11, debe manejar los certificados de notas historias y actuales, de matrícula, de egresado y del pensum, propuestos inicialmente. Y, además, debe permitir la inclusión de nuevos certificados para puedan ser ofertados.



Figura 6-11: Diseño arquitectónico de certificados, diagrama inicial.

Los certificados enumerados y los futuros deben obtener sus datos de los servicios web. Por ello, se requiere de una plataforma de reportes que permita utilizar a los servicios web como fuente de datos y, a su vez, sirva de intermediario para la creación de nuevos certificados. La inclusión de esta plataforma da como resultado el diseño arquitectónico de la Figura 6-12. El componente “Certificados” debe interactuar de la misma manera con esta plataforma, independientemente del certificado que esté siendo procesado.



Figura 6-12: Diseño arquitectónico de Certificados, con la plataforma BIRT.

Como resultado del diseño de la arquitectura del sistema, plasmado en la Figura 6-10, Figura 6-11 y Figura 6-12. El Sistema para kioscos emisores de certificados académicos estaría representado en el diseño arquitectónico de la Figura 6-13. El módulo “certificado” es el que interactúa y provee la funcionalidad al entorno gráfico denominado como “kiosco”. El módulo “certificado” interactúa directamente con los servicios web para realizar la acumulación de la deuda de los certificados, para identificar al usuario y para verificar si el estudiante es egresado o está matriculado. Este módulo, como se mencionó anteriormente, hace uso del módulo “BIRT” para manejar los certificados. Este módulo, a su vez, hace uso de los servicios Web para la obtención de datos.



Figura 6-13: Diseño arquitectónico específico de Sistema de Kioscos emisores de certificados.

“Servicios web” representa a lo implementado mediante el diagrama de clases de la Figura 6-1. “Certificado” simboliza lo implementado mediante las clases Estudiante y Certificado de la Figura 6-2. Y “Kiosco”, representa la interfaz gráfica del kiosco, ubicada en el paquete kiosco de la Figura 6-3. El diseño de esta interfaz gráfica se detalla en la siguiente sección.

6.3 Diseño de la Interfaz gráfica del Kiosco

El diseño de la Interfaz gráfica de usuario (GUI) del Kiosco interactivo, se la realizó en base a las características y los lineamientos presentados en el Capítulo III. Se tomó la recomendación de diseñar una interfaz semejante a un asistente, como el de la Figura 3-1. Se consideró, además, el diseño de una pantalla de bienvenida o principal, que atraiga a los usuarios e indique cual es la funcionalidad del kiosco, tal como se recomienda en la sección 2.5.3 y se puede apreciar en el ciclo de vida de la Figura 2-4.

6.3.1 Pantalla Inicial

Cuando el usuario se acerca al kiosco y toca la pantalla de bienvenida (Figura 6-14), comienza la interacción de la aplicación diseñada como un asistente. En este momento, el kiosco se encuentra en la fase “El usuario interactúa con la aplicación del kiosco” del ciclo de vida de la Figura 2-4.



Figura 6-14: Pantalla Inicial o de Bienvenida

6.3.2 Inicio de Sesión

Como primer paso del asistente, se solicita el ingreso del usuario y contraseña (Figura 6-15), los mismos utilizados en todos los servicios destinados a los estudiantes. El caso del usuario, tiene el formato ua#####, en donde, las letras “ua” vienen escritas por defecto y no pueden ser borradas, esto debido a la recomendación de “reducir la cantidad necesaria de texto a introducir”. Como resultado, solamente es necesario el ingreso de números. Por ello, al estar el cursor ubicado en el campo del usuario, aparecerá un teclado numérico en el formato telefónico (Figura 6-16), lo que cumple con la recomendación de “acomodar el teclado virtual para el ingreso de números”, como se indica en la sección 3.3.5 “Entrada de Datos”.



The screenshot shows the login interface for the 'Sistema de Certificados' at the Universidad del Azuay. At the top left is the university's logo. The main title 'Sistema de Certificados' is centered. Below this, a grey box with a blue header 'Ingrese sus Datos' contains the login form. The 'Usuario' field is pre-filled with 'ua0' and has a cursor at the end. The 'Contraseña' field is empty. To the right of the input fields is a numeric keypad with buttons for digits 1-9, 0, and 'ok', along with a back arrow button. At the bottom left of the page, there is a small icon of a document with a blue arrow pointing to it.

Figura 6-15: Pantalla de Inicio de Sesión.



Figura 6-16: Teclado telefónico.

Cuando se selecciona el campo contraseña o se pulsa el botón OK del teclado de la Figura 6-16, aparecerá en la parte inferior un teclado Alfanumérico, como se indica en la Figura 6-17, este teclado permitirá el correcto ingreso de la contraseña, mediante letras, números y símbolos. Además, el teclado virtual permitirá la eliminación del uso de las teclas especiales, como lo recomienda la sección 2.5.2 “Deshabilitando las teclas especiales”.



Figura 6-17: Pantalla de inicio de sesión al ingresar la contraseña.

En primera instancia, el teclado muestra la distribución QWERTY de números y letras minúsculas, como se indica en la Figura 6-18. Los botones de la derecha, permiten borrar y cambiar la distribución del teclado. Al tocar el botón “ABC”, la distribución cambia a letras mayúsculas, como el teclado de la Figura 6-19. En cambio, si el usuario desea ingresar símbolos que no aparezcan en las otras dos distribuciones, deberá tocar el botón “|@#”, con lo que podrá observar la distribución de la Figura 6-20.

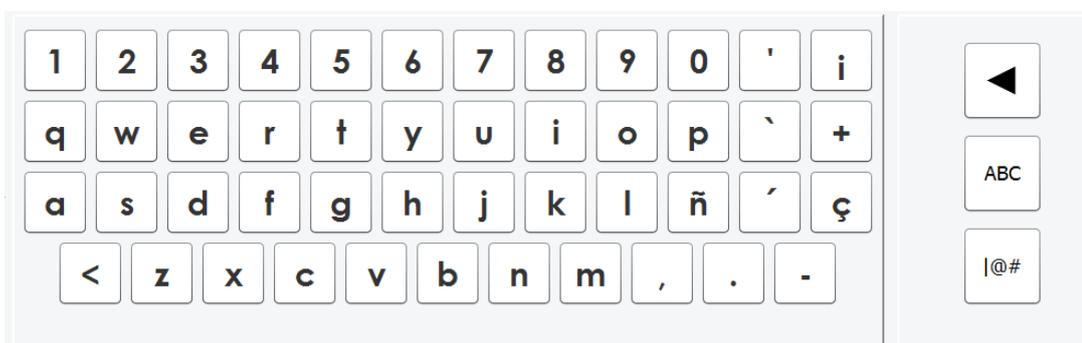


Figura 6-18: Teclado Virtual con distribución QWERTY en minúsculas.

Se eligió un teclado QWERTY porque, como se indicó en la sección 3.1.3 “Teclados Virtuales”, “tiene una mejor eficiencia en usuarios novatos”. Así, al elegir el teclado a utilizar y el tamaño de los botones que representan las teclas, se tomó en cuenta la recomendación de “utilizar un teclado virtual ‘económico’”, que se adapte al espacio disponible en pantalla.



Figura 6-19: Teclado Virtual con distribución QWERTY en mayúsculas.

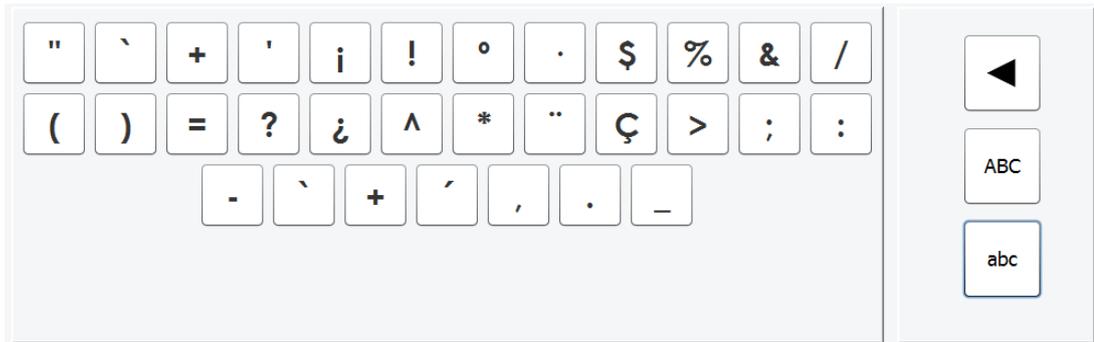


Figura 6-20: Teclado Virtual de símbolos.

6.3.3 Menú de Selección de Certificados

Una vez ingresados el usuario y contraseña, el usuario debe pulsar el botón “Ingresar” que se muestra en la Figura 6-17. Con ello, el asistente cambiará el contenido ubicado en la parte central de la interfaz gráfica, por un menú de selección de certificados, como se muestra en la Figura 6-21.



Figura 6-21: Pantalla con el menú de selección de certificados disponibles.

Este menú con apariencia de botones, fue creado siguiendo los lineamientos de la sección 3.3.6 “Botones y menús”. El botón de salida está claramente diferenciado de los botones del menú de selección de certificados. Así también, estos botones están claramente separados y diferenciados entre ellos, pero, al mismo tiempo, los botones que forman parte del menú de selección de certificados, tienen características comunes, tales como tamaño, color y posición; que les permite ser vistos como parte de un todo.

Estos botones son creados de manera dinámica, dependiendo de los certificados que el módulo de Certificados indique que se encuentran disponibles. Cada botón cuenta con su nombre y una imagen que permita representarlo y distinguirlo de una manera más gráfica del resto. Tanto los botones de este menú, como los otros que se presentan en el resto de la interfaz, cumplen con el lineamiento de un tamaño de 2x2 cm y una separación de 3 mm entre botones, como se indica en la sección 3.3.6.1.

6.3.4 Vista Previa del Certificado

Cuando uno de los certificados es seleccionado, el asistente pasa a la vista previa del certificado (Figura 6-22), que cuenta con el mismo encabezado en la parte superior. En la parte inferior, en cambio, al lado del botón salir, se agrega botón de regresar a la selección de certificados, se muestra un mensaje del costo del certificado, y se agrega el botón de imprimir.

El contenido que se encuentra en la parte central de la pantalla es la vista previa del certificado solicitado. Como algunos certificados poseen más de una página, se utilizaron los lineamientos de la sección 3.3.7 “Desplazamiento y Navegación”, para incluir la barra de navegación de la Figura 6-22. Esta barra permite el desplazamiento o navegación del contenido a la primera página, la anterior, la siguiente y la última página del certificado. Además, mediante el cuadro de texto,

permite conocer qué página del contenido se está mostrando y el número total de páginas del certificado.



Figura 6-22: Pantalla Vista previa del certificado.

Se tomó en cuenta la recomendación de crear una interfaz que solicite confirmación de la selección realizada, cuando las consecuencias sean destructivas o difíciles de deshacer. Por ello, cuando se pulsa el botón imprimir, aparecerá la pregunta de confirmación de la Figura 6-23.

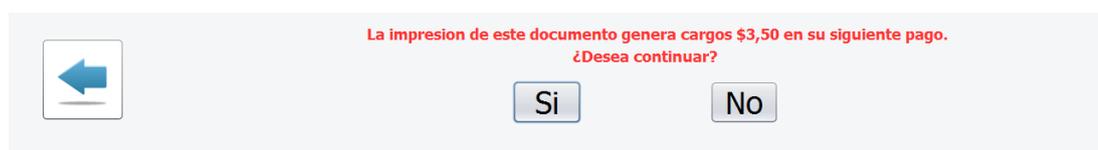


Figura 6-23: Pregunta de confirmación para la impresión del certificado.

6.3.5 Diseño general de la GUI

En el diseño general de la aplicación se consideraron las reglas generales. Se trabajó para que la aplicación sea lo más rápida posible. Se realizó un diseño intuitivo, en el que se guía al usuario a través del proceso de emisión del certificado, limitándole las opciones de selección a aquellas que cada estudiante puede solicitar.

Además, para facilitar la usabilidad de la aplicación, se utiliza únicamente el evento de selección cuando el usuario levanta el dedo de la pantalla. Se obvian las interacciones de arrastrar o sostener. Se prioriza las selecciones de opciones mediante menús, que son las mejores para utilizar en entornos con pantallas táctiles. Se limita el ingreso de valores y texto a los campos usuario y contraseña, minimizando su ingreso con la inclusión del texto predefinido "ua" en el campo usuario. Finalmente, se proporciona una respuesta extra al efecto de presionar un botón, como la escritura del texto en el inicio de sesión, o la pre-visualización del certificado, al presionar el botón del certificado solicitado.

Tomando en cuenta los lineamientos de la sección 3.3.3 "Diseño de la Pantalla", la aplicación fue diseñada para cubrir toda la pantalla, a una resolución de 1280x1024. Además, cuenta con un diseño acorde al resto de interfaces gráficas de las aplicaciones al servicio de los estudiantes, que al mismo tiempo, le permite distinguirse como una máquina interactiva.

Todas las pantallas de la aplicación, desde que inicia la interacción del usuario, cuentan con un diseño consistente, en donde, las áreas funcionales mantienen su ubicación. En la parte superior siempre se ubica el logo de la Universidad del Azuay y el título "Sistema de Certificados". En la parte inferior, se encuentran los botones que indican las opciones de navegación que existen, como el salir, el

regresar y, al final, el imprimir. Cuando se agrega el botón de regresar, el botón salir mantiene su posición.

La parte central es la que concentra la atención e interacción del usuario, y es la que cambia en el transcurso del proceso de solicitud del certificado. Primero contiene el inicio de sesión, que luego es reemplazado por el menú de selección del certificado a emitir y, finalmente, muestra la vista previa del certificado solicitado. Para ejemplificar la experiencia del usuario de una manera similar a la Figura 2-4 y a la Figura 3-1, se ha realizado la Figura 6-24, que representa el flujo de interacción del usuario con la aplicación, tomando en cuenta con los eventos esperados y los opcionales.

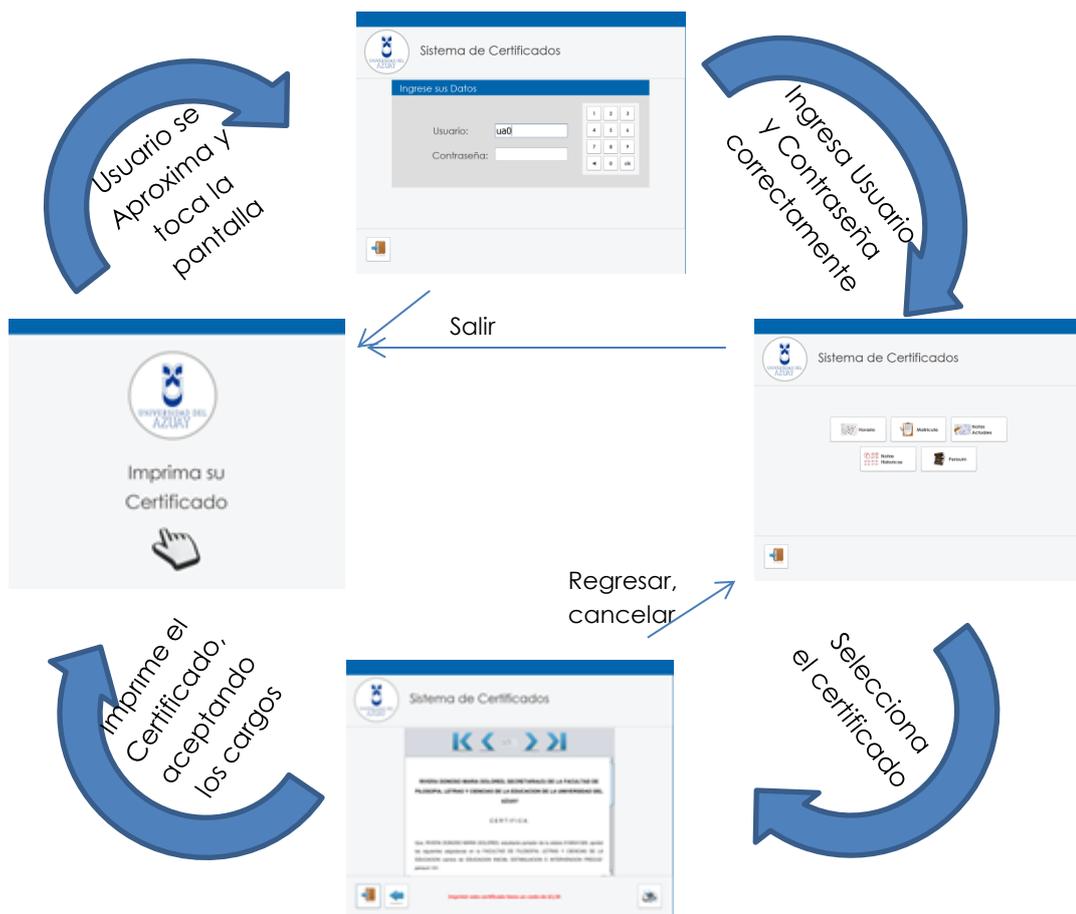


Figura 6-24: Experiencia del Usuario. Flujo de Interacción Usuario – Aplicación.

Con el diseño del sistema, plasmado en los diagramas UML de modelado Estructural, modelado de comportamiento y modelado arquitectónico; se describió la funcionalidad del sistema, su comportamiento y las interacciones con el usuario y con los otros sistemas de la Universidad. El modelado arquitectónico permitió, además, la visualización de los diferentes componentes del sistema y la compatibilidad que se les dio a los mismos. Finalmente, en base a los lineamientos investigados para la creación de interfaces gráficas táctiles para kioscos, se desarrolló el diseño de la Interfaz gráfica de usuario y su secuencia de utilización, de acuerdo a los estándares de interfaces de la Universidad del Azuay.

CAPÍTULO VII

7. DESARROLLO Y PRUEBAS

Tomando en cuenta los objetivos del capítulo I, los conceptos desarrollados en los capítulos II, III y IV, la Especificación de requisitos del capítulo V y, el diseño del sistema y de la interfaz gráfica de usuario de capítulo VI; se llevó a cabo el desarrollo del sistema propuesto y, posteriormente las pruebas del mismo. En este capítulo, se describen los problemas que se encontraron al desarrollar el sistema, junto con las soluciones implementadas. Posteriormente, se describe de manera más detallada la solución al problema de la plataforma de reportes y se justifica la elección. Además, en este capítulo se presenta un pequeño manual para la creación de nuevos reportes con la plataforma escogida. Finalmente, se presenta el plan de pruebas desarrollado para comprobar la funcionalidad del sistema.

7.1 Problemas encontrados y soluciones

7.1.1 Resolución de Pantalla

7.1.1.1 Descripción del Problema

Se requería determinar una resolución de pantalla que permita diseñar de una manera adecuada la interfaz gráfica táctil y sus controles, sin recurrir a altas resoluciones, porque implican tamaños de pantalla muy grandes.

7.1.1.2 Alternativas

Se propusieron dos alternativas: la primera consistía en diseñar la interfaz y sus formularios a una resolución 1024 x 768, con una pantalla de 15". La segunda alternativa, en cambio, utilizaba una resolución de 1280 x 1024, pero requería de una pantalla de 17" como mínimo.

7.1.1.3 Solución

Después de realizar pequeños diseños con la resolución 1024x768, se pudo determinar que los objetos se iban a encontrar muy juntos, e incluso, en algunas ocasiones, el espacio no era suficiente. Por ello, se optó por la resolución de 1280 x 1024, ya que permitía visualizar mejor la información al usuario, y lo ayudaba a desenvolverse de mejor manera en el ambiente táctil del sistema.

7.1.2 Teclado virtual

7.1.2.1 Descripción

Se requería de un teclado virtual incrustado en la aplicación, que permita la introducción de datos en los campos usuario y contraseña, cumpliendo con la recomendación de la sección 3.1.3 “Teclados Virtuales”.

7.1.2.2 Alternativas

Las opciones propuestas eran dos: desarrollar un teclado virtual propio, o encontrar una librería de un teclado virtual gratuito que cumpla con todos nuestros requerimientos.

7.1.2.3 Solución

Se encontraron varios teclados que funcionaban correctamente, pero estos eran diseñados para ambientes web. En cambio, los teclados desarrollados para el ambiente requerido, resultaban inservibles al tratar de utilizarlos. Por lo antes expuesto, se decidió realizar: un teclado numérico, con distribución telefónica, y un teclado alfanumérico, con distribución QWERTY, que permitan ingresar los datos de una manera práctica y sencilla.

7.1.3 Plataforma de Reportes

7.1.3.1 Descripción del Problema

Se requería que el sistema permita al administrador crear reportes sin tener que entrar al módulo de certificados e introducir código. Para ello, se necesitaba de una plataforma de reportes con soporte para servicios web con tecnología SOAP, ya que, los servicios web desarrolladas utilizan esta tecnología.

7.1.3.2 Alternativas

Después de realizar una investigación por internet, se encontraron dos opciones: La primera plataforma de reportes encontrada fue JasperReports o iReports, y la segunda, BIRT (Business Intelligence and Reporting Tools), que es mantenida por la fundación Eclipse.

7.1.3.3 Solución

Después de realizar investigaciones y pruebas con las dos plataformas (iReports y BIRT), se pudo determinar que iReports contaba con un visualizador gráfico SWING, pero no soporta la tecnología SOAP. Por su parte, BIRT brinda este soporte, pero no contaba con un visualizador gráfico para aplicaciones de escritorio. Por ello, luego de la deliberación, descrita a mayor detalle en la sección 7.2.1 de este capítulo, se decidió optar por la plataforma BIRT, que soporta la tecnología SOAP y, por tanto, mediante los servicios web permite crear certificados de una manera más sencilla. Sin embargo, el optar por la plataforma BIRT conllevó a buscar una solución para la vista previa del certificado, lo cual se detalla a continuación.

7.1.4 Vista previa del certificado

7.1.4.1 Descripción del problema

Al solventar el problema anterior, escogiendo la plataforma de reportes BIRT, surgió un nuevo problema: Esta plataforma no cuenta con un visor propio para aplicaciones JAVA de escritorio.

7.1.4.2 Alternativas

En este caso se pudo pensar en tres alternativas: Primero, buscar un visor de reportes BIRT desarrollado por un tercero, lo cual no se encontró. Las otras alternativas eran, en cambio, algo similares entre sí. Estas consistían en exportar el reporte o certificado generado a un formato, en el primer caso a PDF, y en el segundo a HTML. Para el primer caso se requería un lector PDF y, en el segundo, un navegador web.

7.1.4.3 Solución

Debido a que la solución de encontrar un visor para reportes BIRT de un tercero no fue alcanzada, se pasó a las siguientes dos opciones. En cualquiera de las dos, se requería un visualizador del formato exportado. Sin embargo, el incluir un navegador web en la aplicación iba a disminuir su rendimiento. Por lo tanto, se buscó una librería que permita la lectura e interpretación del PDF: la librería PDFRenderer. En base a ella, se construyó el módulo PDFViewer, que además permite la impresión del certificado. Esta solución se encuentra descrita a más detalle en la sección 7.2.2 de este capítulo.

7.1.5 Lenta Apertura de la vista previa del certificado

7.1.5.1 Descripción del Problema

Al momento de seleccionar el certificado, se podía percibir una demora entre 10 y 12 segundos aproximadamente, en generar la vista previa.

7.1.5.2 Alternativas

Después de identificar el problema se encontraron dos alternativas. La primera consistía en colocar una barra de progreso, que se muestre hasta que la vista previa se haya cargado. La otra alternativa, en cambio, consistía en buscar una manera de optimizar el proceso, reducir el tiempo de carga y, así, solucionar el problema.

7.1.5.3 Solución

La mejor solución fue encontrar la razón por la que el proceso tardaba demasiado: la plataforma de reportes requería buscar y cargar las fuentes tipográficas del sistema. Al tiempo de encontrar este inconveniente, se observó que al solicitar un segundo certificado, este tardaba mucho menos tiempo. Por ello, se optó por generar un certificado ficticio al iniciar la aplicación por primera vez (al prender el kiosco). Con ello, se consiguió que todos los certificados sean visualizados en un tiempo promedio de 2 segundos.

7.2 La plataforma de reportes

7.2.1 Elección de la plataforma para los reportes

Para poder llevar a cabo el diseño arquitectónico de la Figura 6-10 y la Figura 6-11, en donde se deben soportar distintos tipos de certificados y estos deben conectarse directamente con los servicios web, fue necesario buscar una

plataforma de reportes que acepte los servicios web como fuente de datos. Las plataformas de reportes más completas y comunes para ser utilizadas con la máquina virtual de JAVA son JasperReports (también conocido como iReport) y Business Intelligence and Reporting Tools (BIRT) de la fundación Eclipse.

Al analizar una comparación³⁷ entre estas dos plataformas, se pudo determinar que, aunque trabajan de distinta manera, soportan los mismos formatos y similares fuentes de datos. Para propósitos del sistema propuesto, se determinó que existían dos características esenciales para determinar la plataforma a ser utilizada. La primera, y más importante, era que la plataforma soporte los servicios web como fuentes de datos y, la segunda, que tenga un visualizador para JAVA SWING, es decir, para una aplicación de escritorio.

En primera instancia, se encontró que ambas plataformas soportaban archivos XML como fuentes de datos (la base para servicios web). Sin embargo, mientras JasperReports (Figura 7-1) sí contaba con un visualizador gráfico del reporte en SWING, BIRT sólo permitía su visualización en HTML. Por lo que, al principio, se optó por JasperReports.

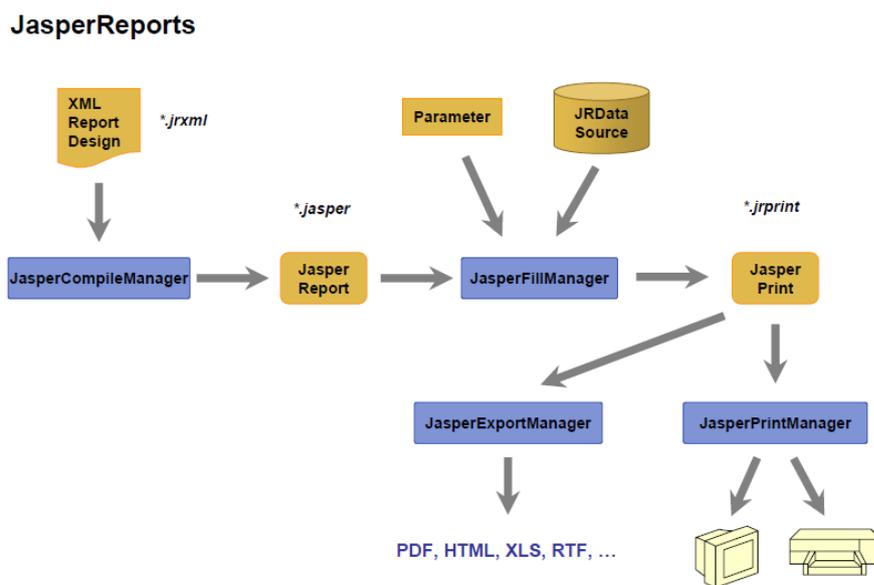


Figura 7-1: Esquema de trabajo de JasperReports.³⁸

³⁷ (Meyer-Otte)

³⁸ Tomado de (Meyer-Otte 11)

Al momento de probar la conectividad de los servicios web con un reporte, se pudo observar que JasperReports permitía el uso de servicios web, pero únicamente aquellos que no eran implementados con tecnología SOAP, que es el caso de servicios web desarrollados, es decir, sólo permitía el uso de archivos XML externos, accedidos mediante una dirección HTTP.

Una opción consistía en, mediante el componente “certificado”, llamar al servicio web y almacenar su respuesta en un archivo XML, que pueda luego ser leído por el reporte diseñado. La solución funcionaba, pero el problema se encontraba en que, con ello, se eliminaba la posibilidad de crear fácilmente nuevos certificados que requieran de datos no proporcionados por los servicios web existentes, es decir, que requieran de nuevos servicios web. Con esta solución, la única manera de incluir nuevos certificados al sistema sería modificando nuevamente el código fuente del componente “certificado”, lo cual, no es el objetivo del sistema propuesto.

Por esta razón, se optó por utilizar la plataforma BIRT, que sí permite el uso de servicios web con tecnología SOAP. El problema a resolver se encontraba en cómo visualizar el contenido del reporte dentro de la aplicación del kiosco.

7.2.2 Uso de la plataforma BIRT

Una vez seleccionada la plataforma BIRT para la realización de los certificados, era necesario buscar la manera de visualizar el reporte generado dentro de la interfaz gráfica. BIRT permite la visualización directa cuando se trata de páginas web, pues genera el reporte en HTML, lo cual, no era útil para la aplicación del kiosco, a menos que se incluyera un navegador web dentro de la aplicación, lo cual, resultaba en un menor rendimiento.

BIRT

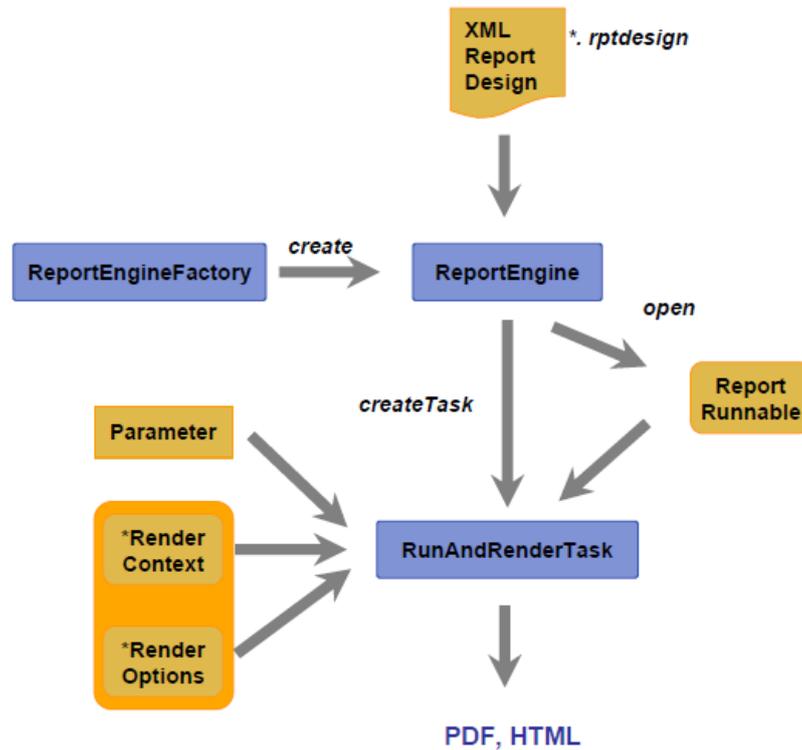


Figura 7-2: Esquema de Trabajo de BIRT³⁹

Se buscaron visualizadores de reportes generados en BIRT pero, a pesar de ser software libre, no se encontró ninguno. Así que, la atención la dirigimos a los formatos de exportación que soporta BIRT (Figura 7-2). El primero es HTML, para su visualización en páginas web, y, el segundo, es PDF. Debido a que la inclusión del navegador web reduciría el rendimiento de la aplicación, se procedió a buscar un visualizador de PDF para JAVA.

Para visualizar el reporte exportado en PDF, únicamente se encontró una librería que **interpretaba** el formato PDF: la librería PDFRenderer. En base a ella, se construyó una librería propia que utiliza estas funciones para plasmar el contenido del PDF en un JPanel. Esta nueva librería también fue complementada con la opción de imprimir el documento y fue nombrada PDFViewer. Esta nueva librería, representada en el diagrama de paquetes de la Figura 6-3, es la utilizada por la

³⁹ Tomado de (Meyer-Otte 12)

interfaz gráfica del kiosco para mostrar la vista previa del certificado. Por lo que, el esquema de los componentes que utiliza la interfaz gráfica del Kiosco sería el de la Figura 7-3.

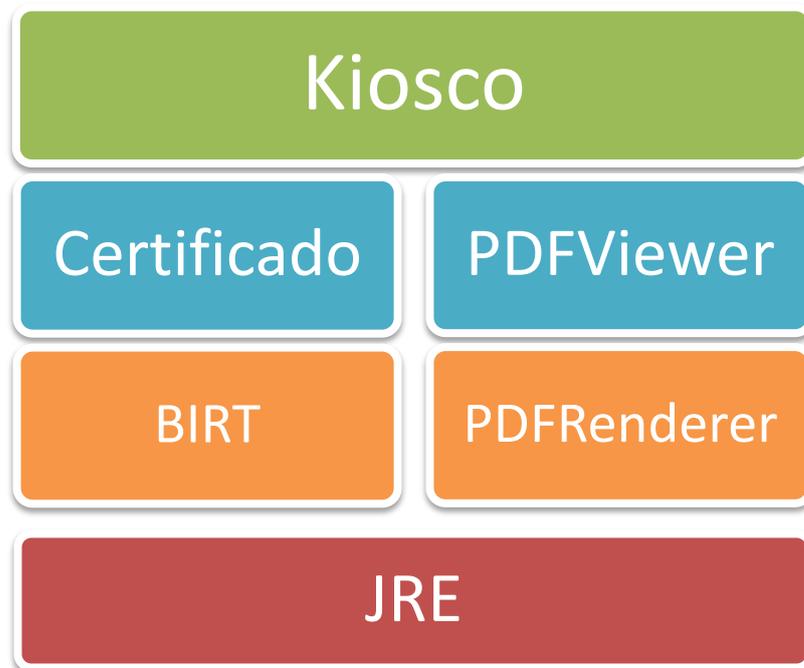


Figura 7-3: Componentes utilizados por la GUI del Kiosco.

7.3 Manual de creación de nuevos certificados

7.3.1 Creación de servicios web

En caso de que se desee crear un certificado académico, que requiera más información de la proporcionada por los servicios web representados en el diagrama de clases de la Figura 6-1, se debe crear un nuevo servicio web. Si el nuevo certificado que se requiere crear puede ser construido en base a la información proporcionada por los servicios web existentes, este paso debe ser obviado.

Este nuevo servicio podrá ser incorporado como un método dentro de las clases `WebServices` o dentro de una nueva clase para servicios web, dependiendo del

sistema del que se requiera obtener la información. En caso de que el nuevo servicio web retorne un valor complejo o compuesto, se requerirá la creación de una nueva clase para la representar los valores a retornar.

7.3.2 Creación del nuevo certificado

7.3.2.1 Nuevo Reporte Tipo Certificado

Primero debemos tener identificados los servicios web que vamos a utilizar, ya sea que existían o fueron creados para el efecto. Con ello en mente, podemos proceder a la creación del nuevo certificado, para ello, accedemos al Entorno de Desarrollo Integrado (IDE) Eclipse with reports.

Dentro de Eclipse seleccionamos File > New > Report. Con lo que podremos observar el asistente de la Figura 7-4, en el que colocamos el nombre del reporte. Este nombre será el que aparezca en el botón que corresponde al certificado, dentro del menú de certificados de la Figura 6-21. Para esta demostración crearemos el Certificado del Pensum que toma el estudiante.

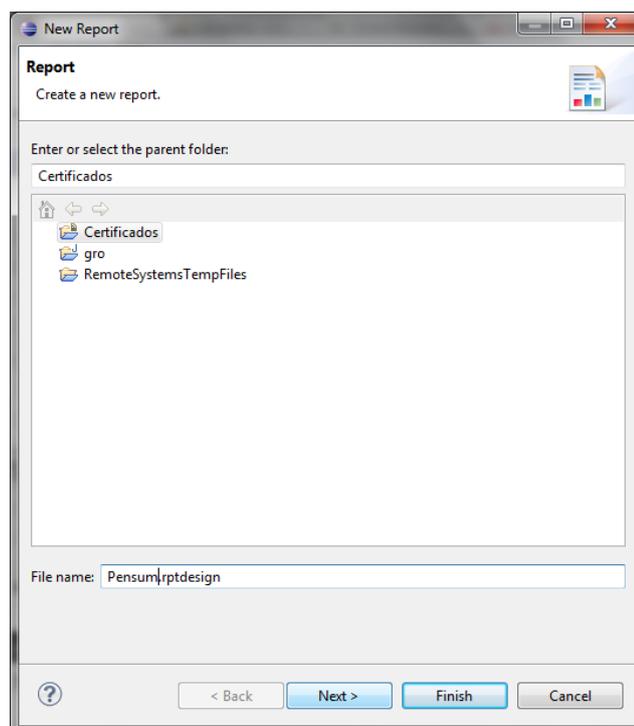


Figura 7-4: Asistente de creación de un nuevo certificado.

Al dar clic en siguiente, el asistente nos solicitará la plantilla desde la cual se quiere crear el nuevo certificado. Seleccionamos la “Plantilla Certificado”, en caso de no aparecer, debe ser importada al proyecto desde la carpeta de certificados. Esta plantilla posee la información básica a ser mostrada en un certificado académico (Figura 7-6), la conexión con los servicios web de créditos, la conexión con el servicio consultarDatosEstudiante y, los parámetros codigo y codigoPeriodo (Figura 7-5), que son los que deben ser enviados por el sistema para la generación del certificado.

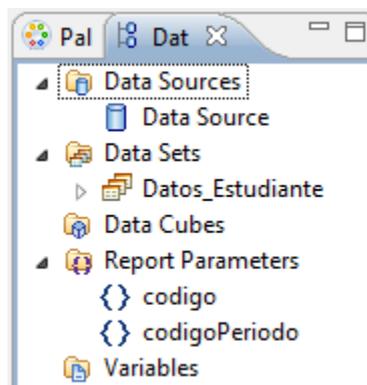


Figura 7-5: Elementos incluidos en la plantilla Certificado.

**SECRETARIA(O) DE LA FACULTAD DE CIENCIA Y
TECNOLOGIA DE LA UNIVERSIDAD DEL AZUAY**

C E R T I F I C A:

Que, **CEDILLO RIVERA, CECILIA JULIA**, estudiante portador de la cédula 0100022605, es estudiante de la FACULTAD DE CIENCIA Y TECNOLOGIA carrera de INGENIERIA ELECTRONICA pensum 111.

Es todo cuanto puedo certificar y autorizo al peticionario dar al presente el uso más conveniente.

Cuenca, Aug 6, 2012 5:48 PM

SECRETARIA(O) DE LA FACULTAD DE CIENCIA Y TECNOLOGIA

Figura 7-6: Vista previa de la plantilla certificado.

Al seleccionar finalizar, se creará el nuevo certificado con la información básica que se muestra en la Figura 7-6.

7.3.2.2 Nueva Fuente de Datos

Ahora, se debe incluir como fuente de datos el o los servicios web que se identificaron o crearon al principio. Para ello, nos dirigimos a la ventana Data Explorer, seleccionamos el elemento Data Set y en el menú desplegable seleccionamos New Data Set, como se muestra en la Figura 7-7.

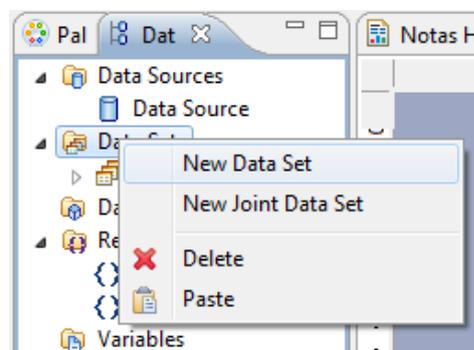


Figura 7-7: Nuevo Data Set

Con ello, se iniciará el asistente para la creación de nuevos conjuntos de datos de la Figura 7-8. En él, debemos seleccionar “Data Source” y dar el nombre del conjunto de datos.

En el siguiente paso del asistente, se debe desplegar el árbol de la fuente de datos, que son los servicios web, y seleccionar el servicio web que se requiere. En este caso seleccionamos el servicio consultarPensum, como se muestra en la Figura 7-9.

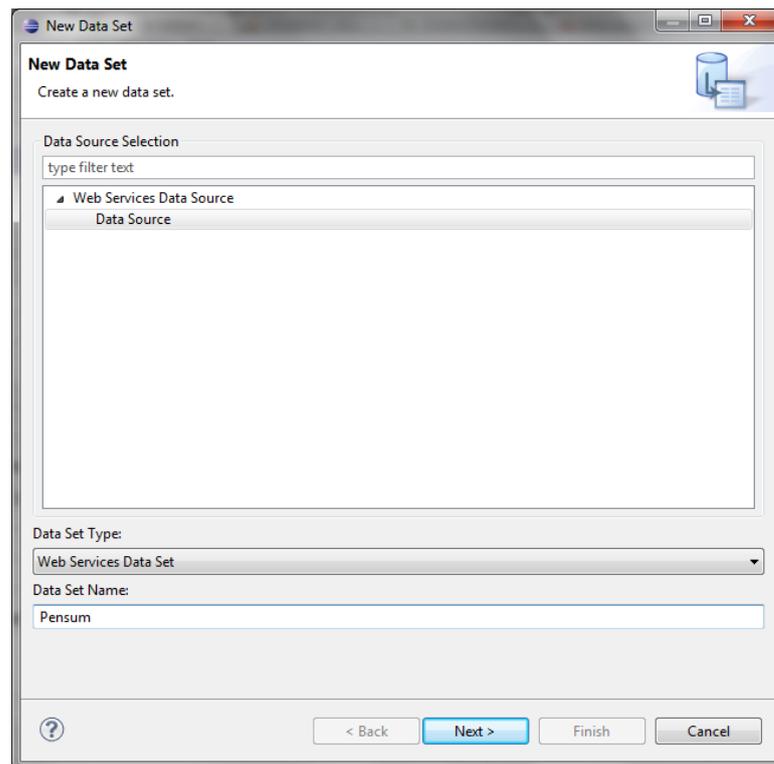


Figura 7-8: Asistente para crear nuevos Data Set.

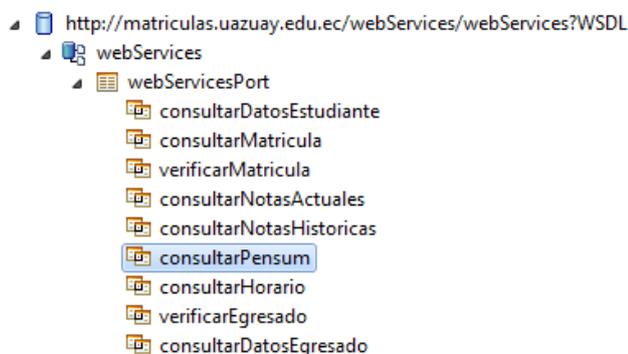


Figura 7-9: Seleccionar el servicio web dentro del WSDL.

La siguiente pantalla nos aparecerá vacía, así que, continuamos con la siguiente, en la que nos solicitará la introducción del SOAP request, es decir, el XML que será enviado para obtener la respuesta del servicio web. El asistente proporciona un esquema ya definido que permite construir este XML, pero hace falta adaptarlo al servicio web implementado. Para conocer la estructura del SOAP request requerido, se debe acceder, desde el navegador web, al tester del servicio web. La

parte que necesita ser cambiada es la que llama al servicio web, que, de una manera inicial, se encuentra de la manera:

```
<m:consultarPensum >  
</m:consultarPensum>
```

Esta debe ser modificada incluyendo el namespace ns2="http://services.creditos.web.uda/" y los parámetros que deben ser enviados al servicio web. En el caso de consultarPensum, sólo se envía el código, por lo que, el esquema anterior debería quedar de la siguiente manera:

```
<ns2:consultarPensum xmlns:ns2="http://services.creditos.web.uda/">  
    <codigo>&?codigo?&</codigo>  
</ns2:consultarPensum>
```

El SOAP Request completo para la llamada al servicio consultarPensum sería el indicado en la Figura 7-10.

```
<?xml version="1.0"?>  
<SOAP-ENV:Envelope  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">  
  <SOAP-ENV:Body>  
    <ns2:consultarPensum xmlns:ns2="http://services.creditos.web.uda/">  
      <codigo>&?codigo?&</codigo>  
    </ns2:consultarPensum>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

Figura 7-10: SOAP Request para consultarPensum.

El contenido "&?codigo?&" dentro de la etiqueta "codigo", representa a un parámetro que permite indicar el código del estudiante. Este parámetro debe tener un valor por defecto para poder configurar el esquema de respuesta. Para ello, seleccionamos "Edit Parameter", después, seleccionamos el parámetro, damos clic en "Edit" e introducimos un parámetro válido en el campo "Default Value", como se muestra en la Figura 7-11.

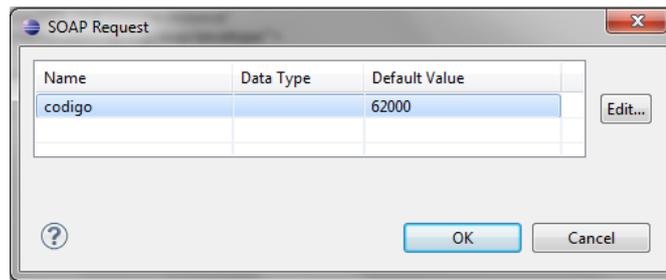


Figura 7-11: Configuración de Parámetros.

Regresando al asistente podemos continuar a la siguiente ventana, en donde, seleccionaremos “Use schema from response” y damos clic en siguiente. En el campo de la derecha “XPath Expression” introducimos “//return”, como se muestra en la Figura 7-12. Esta es una expresión en el lenguaje de consulta XML conocido como XPath, que permite obtener todo el contenido de la etiqueta “return” sin importar dentro de que otra etiqueta se encuentre. Los servicios web empaquetan toda la respuesta dentro de esta etiqueta “return”.

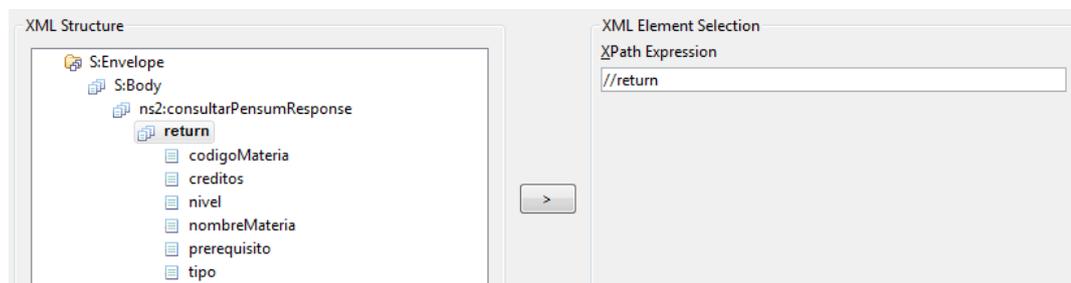


Figura 7-12: Selección de la expresión de consulta XPath.

En el siguiente paso, el asistente nos solicita seleccionar los elementos que van a componer el data set. Esto es lo homologa a los campos que se seleccionan en un Select de una tabla, en el lenguaje SQL. Para ello, nos ubicamos en la parte izquierda y pasamos los campos a utilizar a la parte derecha mediante el botón “>”. Posteriormente, se debe seleccionar cada campo de la parte derecha y editar su tipo de dato mediante el botón “Edit”. Con ello obtendremos el resultado de la Figura 7-13.

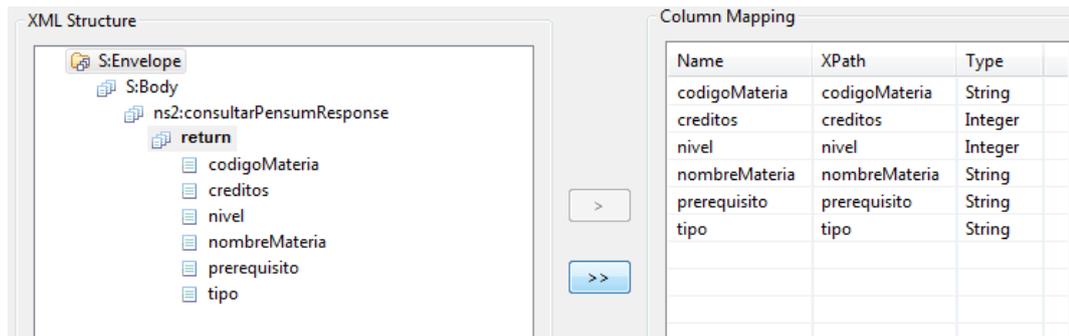


Figura 7-13: Selección y configuración de elementos a formar parte del Data set.

Una vez seleccionados y configurados todos los elementos, seleccionamos finalizar. Con ello, aparecerá un nuevo recuadro. En él, nos dirigimos a “Parameters”. Estos parámetros son los que fueron incluidos en el SOAP Request de la Figura 7-10 y deben ser asociados con algún parámetro del reporte para obtener de él, el valor requerido. Para esto, en cada parámetro seleccionamos “Edit”, con lo que obtendremos la ventana de la Figura 7-14. En la parte “Linked to report parameter” seleccionamos el parámetro al que va a estar vinculado y seleccionamos OK, con lo que la configuración del Data set estaría terminada.

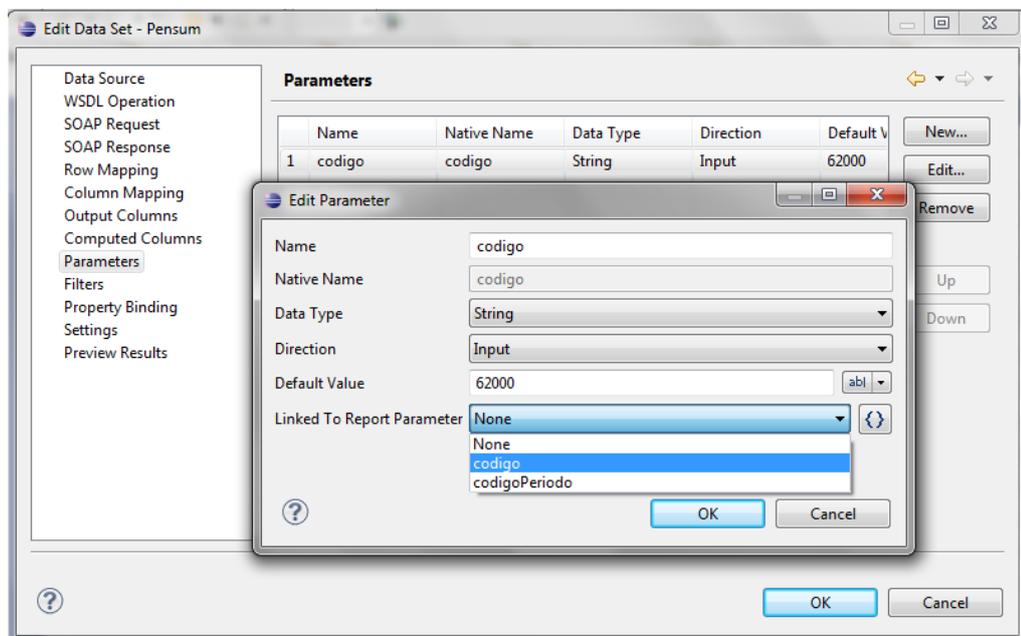


Figura 7-14: Crear un enlace entre el parámetro del reporte y el parámetro del data set.

7.3.2.3 Tamaño de Hoja

Para configurar el tamaño de hoja del certificado, nos ubicamos en la parte inferior del reporte y seleccionamos “Master Page”, después, nos vamos a la ventana de propiedades y buscamos “Type” en la que debemos seleccionar, típicamente, A4, como se muestra en la Figura 7-15.

Style	
Text	
Type	A4
Width	210mm : Inherited

Figura 7-15: Selección del tamaño de hoja en el certificado.

7.3.2.4 Inclusión de nuevos datos en el certificado

Para introducir nuevos datos desde los data set, basta con arrastrar el campo hacia el reporte. Si se requiere crear una tabla con varios campos del data set, como es el caso de las materias del certificado de pensum, se arrastra todo el data set al área de diseño del reporte. Lo que nos dará como resultado la tabla de la Figura 7-16.

codigoMateria	nombreMateria	nivel	creditos	prerequisito	tipo
[codigoMateria]	[nombreMateria]	[nivel]	[creditos]	[prerequisito]	[tipo]
Footer Row					

Figura 7-16: Tabla del Data set Pensum

En caso de que se requiera realizar una agrupación de datos por una columna, como en el caso del certificado de pensum, en el que se debe agrupar las materias por el nivel en el que se encuentran; debemos seleccionar con clic derecho la tabla y en el menú desplegable seleccionar “Insert Group”, como se indica en la Figura 7-17.

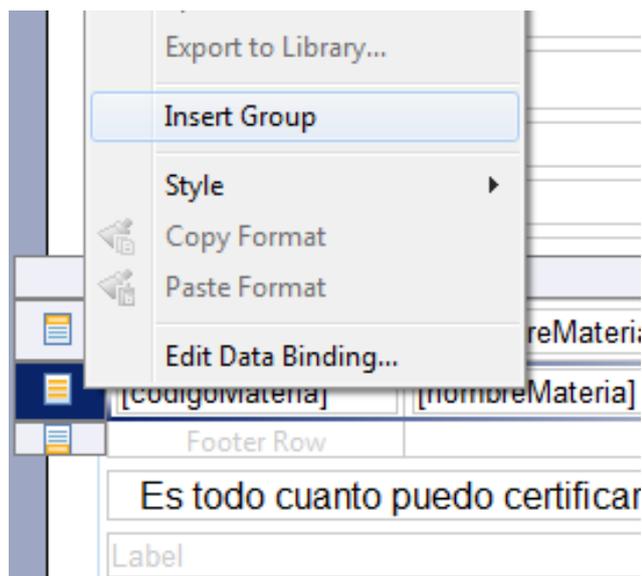


Figura 7-17: Insertar nueva forma de agrupar datos.

A continuación, en la ventana que aparece (Figura 7-18), introducimos el nombre del Grupo y, en “Group On”, seleccionamos la columna por la cual se deben agrupar los datos. En el caso del certificado de pensum, la columna que permite la agrupación sería nivel, que representa el ciclo al que corresponde la materia.

Name: Ciclos

Group On: nivel

Interval: No Interval

Range:

Use fixed base value for interval:

Hide Detail

Figura 7-18: Creación de un nuevo grupo.

En caso de que se requiera filtrar los datos que se desean mostrar, en la parte inferior de la ventana existe una tabla que permite el ingreso de filtros y la manera de ordenar los datos, esta sección es la representada en la Figura 7-19. En el caso del certificado de pensum, se aplicó un filtro en el que el nivel debe ser mayor a 0, pues, en el nivel -1 se encuentran los créditos extras y en el 0, los créditos optativos. Estos niveles deben ser presentados de manera separada.

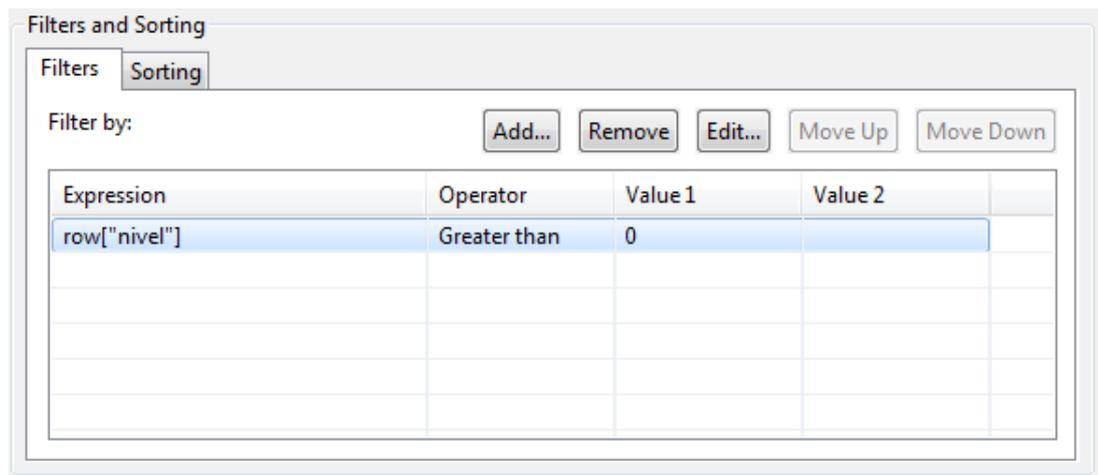


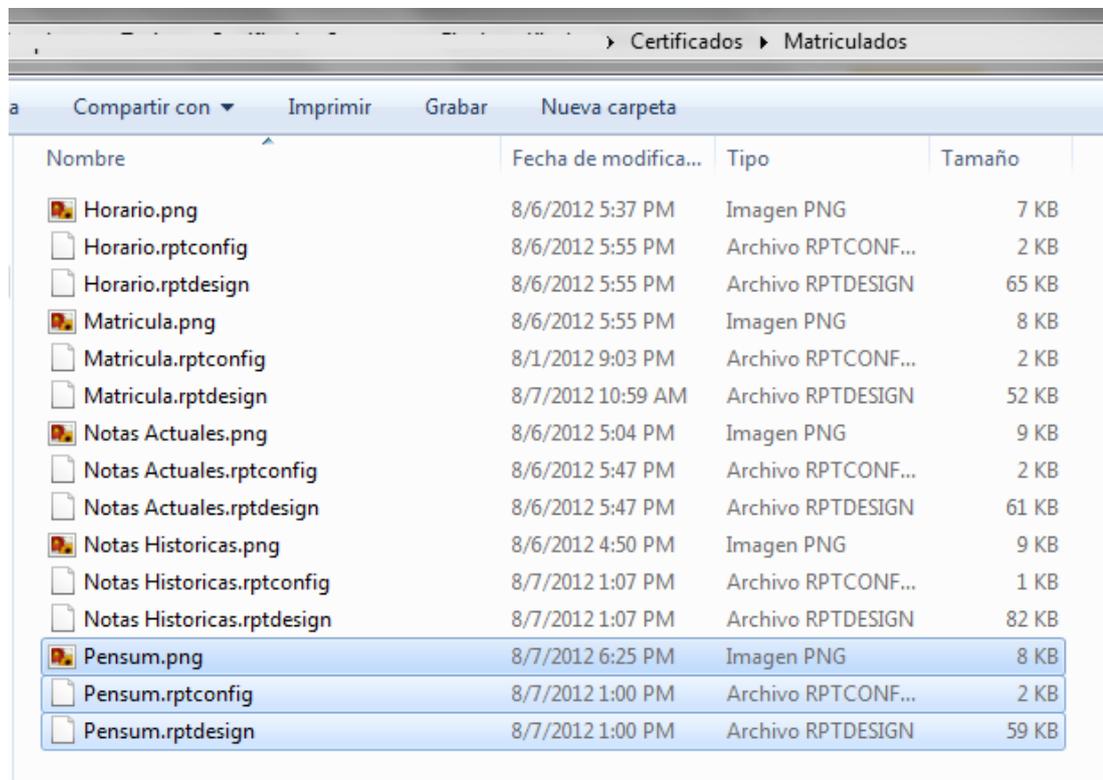
Figura 7-19: Filtros para los datos mostrados en el grupo.

Finalmente, se deben realizar los cambios al contenido y al aspecto del certificado. Esto se logra de manera similar a cualquier otro entorno gráfico: mediante la selección de los elementos y uso de los atributos de la ventana propiedades. Con ello, el diseño final del certificado podría ser como el de la Figura 7-20.

7.3.3 Integración del certificado al sistema

El sistema de certificados conoce cuales son los certificados que están disponibles para los estudiantes, mediante la lectura de los elementos de dos carpetas que se encuentran en la carpeta Certificados, dentro del directorio de ejecución de la aplicación (Figura 7-21). En la carpeta Matriculados, se deben encontrar todos los archivos de los certificados que puedan ser solicitados por estudiantes que se encuentren matriculados al momento de solicitar el certificado. Análogamente, en la carpeta Egresados, se encuentran los certificados que pueden ser solicitados por estudiantes egresados.

Dependiendo del caso, el sistema hace lectura de los archivos de la carpeta que terminan en “.rtpdesign”, y extrae su nombre. Este nombre será el presentado en el menú de selección de certificados de la Figura 6-21, junto con la imagen que contenga el mismo nombre y la extensión “.png”.



Nombre	Fecha de modifica...	Tipo	Tamaño
 Horario.png	8/6/2012 5:37 PM	Imagen PNG	7 KB
 Horario.rptconfig	8/6/2012 5:55 PM	Archivo RPTCONF...	2 KB
 Horario.rtpdesign	8/6/2012 5:55 PM	Archivo RPTDESIGN	65 KB
 Matricula.png	8/6/2012 5:55 PM	Imagen PNG	8 KB
 Matricula.rptconfig	8/1/2012 9:03 PM	Archivo RPTCONF...	2 KB
 Matricula.rtpdesign	8/7/2012 10:59 AM	Archivo RPTDESIGN	52 KB
 Notas Actuales.png	8/6/2012 5:04 PM	Imagen PNG	9 KB
 Notas Actuales.rptconfig	8/6/2012 5:47 PM	Archivo RPTCONF...	2 KB
 Notas Actuales.rtpdesign	8/6/2012 5:47 PM	Archivo RPTDESIGN	61 KB
 Notas Historicas.png	8/6/2012 4:50 PM	Imagen PNG	9 KB
 Notas Historicas.rptconfig	8/7/2012 1:07 PM	Archivo RPTCONF...	1 KB
 Notas Historicas.rtpdesign	8/7/2012 1:07 PM	Archivo RPTDESIGN	82 KB
 Pensum.png	8/7/2012 6:25 PM	Imagen PNG	8 KB
 Pensum.rptconfig	8/7/2012 1:00 PM	Archivo RPTCONF...	2 KB
 Pensum.rtpdesign	8/7/2012 1:00 PM	Archivo RPTDESIGN	59 KB

Figura 7-21: Inclusión del nuevo certificado en el directorio.

Para cumplir con esto, el certificado diseñado y guardado, debe ser incluido en la carpeta correspondiente. Para lo cual, copiamos los archivos “.rptdesign” y “.rtpconfig” generados, dentro de la carpeta Matriculados o Egresados, dependiendo del caso. Por ejemplo, el certificado de pensum está disponible únicamente para estudiantes matriculados, por lo que será colocado en esta carpeta, como se muestra en la Figura 7-21.

Como se mencionó anteriormente, el sistema permite la inclusión de una imagen que será la que aparezca junto al nombre del certificado, en la selección de certificados de la Figura 6-21. Esta imagen debe tener el mismo nombre que los archivos “.rptdesign” y “.rtpconfig”, y debe estar guardada en formato PNG, como lo indica la Figura 7-21.

7.4 Plan de Pruebas

7.4.1 Pruebas de Servicios Web

7.4.1.1 Web Service consultarDatosEstudiante

Campo	Entradas	Salidas
Código	1234	Null. Excepción: Exception
Código	62000	<return> <carrera>INGENIERIA ELECTRONICA</carrera> <codigo>62000</codigo> <codigoCarrera>IE1</codigoCarrera> <codigoPeriodo>78</codigoPeriodo> <facultad> FACULTAD DE CIENCIA Y TECNOLOGIA </facultad> <identificacion>0104xxxxxxx</identificacion> <nombre>XXXX XXXX XXXX</nombre> <nombreSecretaria>Xxxxx Xxxxx</nombreSecretaria> <pensum>111</pensum> <periodoActual>

		PERIODO SEP/2012 - FEB/2013 </periodoActual> </return>
--	--	--

7.4.1.2 Web Service consultarMatricula

Campo	Entradas	Salidas
Código	1234	[]
Periodo	-1	
Código	62000	[]
Periodo	-1	
Código	1234	[]
Periodo	66	
Código	62000	<return>
Periodo	66	<codigoMateria>CTE0241</codigoMateria> <creditos>4</creditos> <nivel>0</nivel> <nombreMateria> QUÍMICA GENERAL </nombreMateria> <paralelo>D</paralelo> <porcentajeAsistencia> 93.75 </porcentajeAsistencia> </return> <return> <codigoMateria>CTE0146</codigoMateria> <creditos>4</creditos> <nivel>0</nivel> <nombreMateria> INFORMÁTICA I PARA IE1 </nombreMateria> <paralelo>D</paralelo> <porcentajeAsistencia> 84.375 </porcentajeAsistencia> </return> ...

7.4.1.3 Web Service verificarMatricula

Campo	Entradas	Salidas
Código	1234	False.
Código	62000	True.

7.4.1.4 Web Service consultarNotasActuales

Campo	Entradas	Salidas
Código	1234	[]
Periodo	-1	
Código	62000	[]
Periodo	-1	
Código	1234	[]
Periodo	66	
Código	62000	<return>
Periodo	66	<codigoMateria>CTE0146</codigoMateria> <creditos>4</creditos> <nivel>1</nivel> <nombreMateria> INFORMÁTICA I PARA IE1 </nombreMateria> <condicion>A</condicion> <notaAporte>27.1</notaAporte> <notaExamen>15.0</notaExamen> <notaFinal>42.1</notaFinal> <notaSuspenso>0.0</notaSuspenso> </return> <return> <codigoMateria>CTE0241</codigoMateria> <creditos>4</creditos> <nivel>1</nivel> <nombreMateria> QUÍMICA GENERAL </nombreMateria> <condicion>A</condicion> <notaAporte>24.0</notaAporte>

		<pre> <notaExamen>10.0</notaExamen> <notaFinal>34.0</notaFinal> <notaSuspenso>0.0</notaSuspenso> </return> ... </pre>
--	--	---

7.4.1.5 Web Service consultarNotasHistoricas

Campo	Entradas	Salidas
Código	1234	[].
Código	62000	<pre> <return> <codigoMateria>CTE0110</codigoMateria> <creditos>6</creditos> <nivel>1</nivel> <nombreMateria>FÍSICA I</nombreMateria> <comentario> CONVALIDADA, 1ra Matricula </comentario> <condicion>A</condicion> <notaFinal>32.5</notaFinal> <periodoTomado> PERIODO SEP/2010 - FEB/2011 </periodoTomado> </return> <return> <codigoMateria>CTE0118</codigoMateria> <creditos>6</creditos> <nivel>1</nivel> <nombreMateria> GEOMETRÍA Y TRIGONOMETRÍA </nombreMateria> <comentario> CONVALIDADA, 1ra Matricula </comentario> <condicion>A</condicion> <notaFinal>33.5</notaFinal> <periodoTomado> PERIODO SEP/2010 - FEB/2011 </pre>

```

        </periodoTomado>
    </return>
    ...

```

7.4.1.6 Web Service consultarPensum

Campo	Entradas	Salidas
Código	1234	[]
Código	62000	<pre> <return> <codigoMateria>CTE0110</codigoMateria> <creditos>6</creditos> <nivel>1</nivel> <nombreMateria>FÍSICA I</nombreMateria> <prerequisito/> <tipo>REQUERIDA</tipo> </return> <return> <codigoMateria>CTE0118</codigoMateria> <creditos>6</creditos> <nivel>1</nivel> <nombreMateria> GEOMETRÍA Y TRIGONOMETRÍA </nombreMateria> <prerequisito/> <tipo>REQUERIDA</tipo> </return> <return> <codigoMateria>CTE0146</codigoMateria> <creditos>4</creditos> <nivel>1</nivel> <nombreMateria> INFORMÁTICA I PARA IE1 </nombreMateria> <prerequisito/> <tipo>REQUERIDA</tipo> </return> <return> <codigoMateria>CTE0183</codigoMateria> </pre>

		<pre> <creditos>6</creditos> <nivel>1</nivel> <nombreMateria> MATEMÁTICAS I </nombreMateria> <prerequisito/> <tipo>REQUERIDA</tipo> </return> ... </pre>
--	--	--

7.4.1.7 Web Service consultarHorario

Campo	Entradas	Salidas
Código	1234	[]
Periodo	-1	
Código	62000	[]
Periodo	-1	
Código	1234	[]
Periodo	66	
Código	62000	<return>
Periodo	66	<pre> <dia>Lunes</dia> <horaFin>17</horaFin> <horaInicio>16</horaInicio> <minutoDesde>0</minutoDesde> <minutoFin>0</minutoFin> <nombreMateria> QUÍMICA GENERAL </nombreMateria> </return> <return> <dia>Lunes</dia> <horaFin>18</horaFin> <horaInicio>17</horaInicio> <minutoDesde>0</minutoDesde> <minutoFin>0</minutoFin> <nombreMateria> QUÍMICA GENERAL </pre>

```

        </nombreMateria>
    </return>

    <return>
        <dia>Martes</dia>
        <horaFin>19</horaFin>
        <horaInicio>18</horaInicio>
        <minutoDesde>0</minutoDesde>
        <minutoFin>0</minutoFin>
        <nombreMateria>
            INFORMÁTICA I PARA IE1
        </nombreMateria>
    </return>
    ...

```

7.4.1.8 Web Service verificarEgresado

Campo	Entradas	Salidas
Código	62000	False.
Código	(estudiante egresado cod>60000)	True.

7.4.1.9 Web Service consultarDatosEgresado

Campo	Entradas	Salidas
Código	1234	Null.
Código	62000	<return> <calificacionTotal>37.13</calificacionTotal> <creditosExtras>2</creditosExtras> <creditosOptativos>0</creditosOptativos> <creditosRequeridos>102</creditosRequeridos> <equivalencia>Buena</equivalencia> <fechaEgreso> 2012-07-28T00:00:00-05:00 </fechaEgreso> </return>

7.4.2 Pruebas de Sesión, interfaz gráfica y de certificados

7.4.2.1 Ingreso al sistema

Campo	Entradas	Salidas
Usuario Contraseña	ua043330 XXXXXXXX	Error: El usuario que ingreso no está habilitado, sólo usuarios > 60000, por favor acérquese a secretaria. Figura 7-22
Usuario Contraseña	ua060924 (usuario no matriculado) contraseña correcta	Error: El usuario que ingreso no se encuentra matriculado o egresado, por favor intente de nuevo. Figura 7-23
Usuario Contraseña	ua187856 XXXXXXXX	Error: El usuario o contraseña ingresados no son válidos, por favor intente de nuevo. Figura 7-24
Usuario Contraseña	ua060896 123	Error: La contraseña debe ser mayor a 8 caracteres, por favor intente de nuevo. Figura 7-25
Usuario Contraseña	ua060896 Contraseña incorrecta	Error: El usuario o contraseña ingresados no son válidos, por favor intente de nuevo. Figura 7-24
Usuario Contraseña	ua060896 [vacía]	Error: Por favor ingrese su contraseña. Figura 7-26
Usuario Contraseña	ua0XXXXX (egresado) Contraseña correcta	Lista de Certificados Certificado de egresado.
Usuario Contraseña	ua060896 (matriculado) Contraseña correcta	Lista de Certificados (Figura 6-21) Certificado de Notas Actuales. Certificado de Notas históricas. Certificado del horario de clases. Certificado del <i>pensum</i> de la carrera. Certificado de matrícula.

Ingrese sus Datos

Usuario:

Contraseña:

1	2	3
4	5	6
7	8	9
◀	0	ok

El usuario que ingreso no está habilitado, sólo usuarios > 60000, por favor acérquese a secretaria.

Figura 7-22: Error: usuario no habilitado

Ingrese sus Datos

Usuario:

Contraseña:

1	2	3
4	5	6
7	8	9
◀	0	ok

El usuario que ingreso no se encuentra matriculado o egresado, por favor intente de nuevo

Figura 7-23: Error: Estudiante no matriculado o no está egresado.

Ingrese sus Datos

Usuario:

Contraseña:

1	2	3
4	5	6
7	8	9
◀	0	ok

El usuario o contraseña ingresados no son válidos, por favor intente de nuevo.

Figura 7-24: Error: Usuario o contraseña no coinciden con un estudiante.

Ingrese sus Datos

Usuario:

Contraseña:

La contraseña debe ser mayor a 8 caracteres, por favor intente de nuevo

Ingresar

Figura 7-25: Error: contraseña de menor longitud a la permitida.

Ingrese sus Datos

Usuario:

Contraseña:

Por favor ingrese su contraseña

Ingresar

Figura 7-26: Error: no se proporciona contraseña.

7.4.2.2 Selección del Certificado de Notas Actuales

Campo	Entradas	Salidas
Certificado	Selección del certificado de notas actuales, con un estudiante matriculado.	Vista previa del Certificado de acuerdo al ANEXO 6: CERTIFICADO DE NOTAS DEL CICLO ACTUAL. Mensaje: La impresión de este documento genera cargos de \$3.50. ¿Desea continuar? Figura 6-23
Opción Si		Regreso a la vista previa.
Opción No	NO	

Opción Si	SI	Orden de Impresión del Certificado, Acumulación del valor en la cuenta del estudiante.
Opción No		Pre visualización cerrada. Sesión cerrada. Regresa a pantalla de Bienvenida.

7.4.2.3 Selección del Certificado de Horario

Campo	Entradas	Salidas
Certificado	Selección del certificado de horario, con un estudiante matriculado.	Vista previa del Certificado de acuerdo al ANEXO 5: CERTIFICADO CON EL HORARIO. Mensaje: La impresión de este documento genera cargos de \$3.50. ¿Desea continuar? Figura 6-23
Opción Si		Regreso a la vista previa.
Opción No	NO	
Opción Si	SI	Orden de Impresión del Certificado, Acumulación del valor en la cuenta del estudiante.
Opción No		Pre visualización cerrada. Sesión cerrada. Regresa a pantalla de Bienvenida.

7.4.2.4 Selección del Certificado de Matricula

Campo	Entradas	Salidas
Certificado	Selección del certificado de matrícula, con un estudiante matriculado.	Vista previa del Certificado de acuerdo al ANEXO 1: CERTIFICADO DE MATRICULA. Mensaje: La impresión de este documento genera cargos de \$3.50. ¿Desea continuar? Figura 6-23
Opción Si		Regreso a la vista previa.
Opción No	NO	

Opción Si	SI	Orden de Impresión del Certificado, Acumulación del valor en la cuenta del estudiante.
Opción No		Pre visualización cerrada. Sesión cerrada. Regresa a pantalla de Bienvenida.

7.4.2.5 Selección del Certificado de Notas Históricas

Campo	Entradas	Salidas
Certificado	Selección del certificado de notas históricas, con un estudiante matriculado.	Vista previa del Certificado de acuerdo al ANEXO 3: CERTIFICADO DE NOTAS HISTÓRICAS. Mensaje: La impresión de este documento genera cargos de \$3.50. ¿Desea continuar? Figura 6-23
Opción Si		Regreso a la vista previa.
Opción No	NO	
Opción Si	SI	Orden de Impresión del Certificado, Acumulación del valor en la cuenta del estudiante.
Opción No		Pre visualización cerrada. Sesión cerrada. Regresa a pantalla de Bienvenida.

7.4.2.6 Selección del Certificado de Pensum

Campo	Entradas	Salidas
Certificado	Selección del certificado de Pensum, con un estudiante matriculado.	Vista previa del Certificado de acuerdo al ANEXO 4: CERTIFICADO CON EL PENSUM. Mensaje: La impresión de este documento genera cargos de \$3.50. ¿Desea continuar? Figura 6-23
Opción Si		Regreso a la vista previa.
Opción No	NO	

Opción Si	SI	Orden de Impresión del Certificado, Acumulación del valor en la cuenta del estudiante. Pre visualización cerrada. Sesión cerrada. Regresa a pantalla de Bienvenida.
Opción No		

7.4.2.7 Selección del Certificado de Egresado

Campo	Entradas	Salidas
Certificado	Selección del certificado de Egresado, con un estudiante egresado.	Vista previa del Certificado de acuerdo al ANEXO 2: CERTIFICADO DE EGRESADO. Mensaje: La impresión de este documento genera cargos de \$3.50. ¿Desea continuar? Figura 6-23
Opción Si	NO	Regreso a la vista previa.
Opción No		
Opción Si	SI	Orden de Impresión del Certificado, Acumulación del valor en la cuenta del estudiante. Pre visualización cerrada. Sesión cerrada. Regresa a pantalla de Bienvenida.
Opción No		

En este capítulo, se dieron a conocer los problemas enfrentados durante el transcurso del proyecto, para alcanzar los requisitos funcionales y no funcionales. Se indicaron las alternativas que se propusieron y la solución que se tomó. El problema de la plataforma de reportes y su elección, se la describió más a fondo por la importancia que esta tiene en el sistema en general. El manual para crear nuevos certificados se lo incluyó como una muestra de lo fácil que es realizar un nuevo certificado. Finalmente, el plan de pruebas presentado fue de mucha utilidad para comprobar el correcto funcionamiento del sistema, y el cumplimiento de los requisitos expuestos en el ERS.

CAPÍTULO VIII

8. CONCLUSIONES Y RECOMENDACIONES

8.1 Conclusiones

Al realizar el análisis, diseño, desarrollo y validación del Sistema informático para kioscos emisores de certificados académicos para la Universidad del Azuay, se cumplieron los objetivos propuestos en esta tesis y, además, se pudo concluir que:

En el desarrollo de una solución informática, no sólo se debe tomar en cuenta las necesidades de los usuarios, sino también los recursos físicos sobre los que se debe implementar el sistema. Así, en el desarrollo de esta tesis, fue necesario el buscar información sobre los kioscos interactivos, para conocer sus características y proponer un “kiosco requerido para el proyecto”, que, a su vez, permita diseñar el sistema en base a él. El optar por una pantalla táctil, también implicó conocer más a fondo las características y limitaciones de estas pantallas, así como, los lineamientos que se recomiendan tomar en cuenta para el diseño de interfaces gráficas táctiles. Estos lineamientos, junto con el tamaño y resolución de pantalla propuestos, sirvieron de base para dimensionar y diseñar adecuadamente la interfaz gráfica.

Los servicios web o web services, junto con la arquitectura orientada a servicios SOA, constituyen una herramienta poderosa para implementar sistemas de integración empresarial, y, en otros casos, para brindar servicios a otras instituciones de una manera estandarizada, gracias al formato XML. Así, en la Universidad del Azuay, los servicios web quedarán disponibles para ser utilizados por esta y otras aplicaciones que los requieran, formando un middleware entre estas aplicaciones. Además, estos servicios web, facilitan la creación de nuevos certificados con datos adicionales a los proporcionados por los servicios web

existentes, pues solo se requieren la creación de un nuevo servicio, sin modificar el resto del sistema de certificados.

La realización de una Especificación de Requisitos de Software, que agrupe los requisitos por medio de casos de uso, es una técnica que permite realizar un análisis adecuado del sistema que se desea construir. Así también, mediante la revisión conjunta con las partes interesadas, permite llegar a consensos y clarificaciones que ayudan a que el sistema se desarrolle de una manera fluida y sin mayores modificaciones que retrasen el proyecto. Además, con los requerimientos no funcionales detallados en el ERS, se puede vigilar constantemente que el sistema sea de calidad y cumpla con restricciones de velocidad, confiabilidad, etc. Como resultado de todo el ERS, se tiene un análisis completo del sistema, en el que se indica: qué debe hacer el sistema (R. funcionales p. ej. Imprimir el certificado) y qué es el sistema (R. no funcionales p. ej. Seguro, confiable, etc.).

UML permiten realizar un análisis y diseño orientado a objetos mediante diagramas fáciles de comprender y entender, tanto para las personas que desarrollan el sistema y para las partes interesadas. Los diagramas UML, además, permiten nutrir al análisis y diseño desde distintas perspectivas, ayudando a que ningún detalle del sistema sea obviado. Estos diagramas permiten representar la parte estática o estructural del sistema, su parte dinámica o de comportamiento y su modelo arquitectónico general. Como resultado, se obtiene un análisis y diseño sólidos, que sirven de base para el desarrollo adecuado y ordenado del sistema.

Los sistemas no deben basarse únicamente en la construcción de software desde cero. En la actualidad, existen diferentes componentes, gratuitos y pagados, que brindan funcionalidades probadas, lo que puede disminuir el tiempo de desarrollo del sistema o, incluso, dar solución a un problema encontrado. Por ejemplo, la

plataforma BIRT es una herramienta de gran utilidad para sistemas que requieran de reportes o documentos similares, por lo que fue utilizada para la generación de los certificados. Así también, la librería PDFRenderer permitió solventar el problema de visualizar el certificado solicitado, sin la necesidad de aumentar el tiempo de desarrollo.

Finalmente, podemos concluir que el realizar un plan de pruebas ayuda a realizar un control efectivo de los problemas que pueden darse, y de los errores que deben ser controlados y notificados. Además, permite verificar la funcionalidad del sistema desarrollado, así como su calidad. Cumpliendo, de esta manera, con el objetivo de validar que el software brinde el servicio propuesto.

8.2 Recomendaciones

Después de haber realizado un estudio de algunos de los sistemas y procesos de la Universidad, y con de la realización del sistema propuesto, podemos recomendar que:

Al implementar el sistema emisor de certificados académicos, se deberá comprar el o los kioscos, tomando en cuenta las características presentadas en la sección 2.6 “Especificadores del Kiosco requerido para el proyecto”, especialmente en lo relacionado con la resolución de la pantalla táctil, y que el kiosco no cuente con un teclado y ratón, para así brindar una mayor seguridad.

Para no realizar un gasto innecesario, se recomienda la modificar el reglamento de certificados, para evitar la instalación de un kiosco emisor de certificados por cada una de las facultades de la Universidad del Azuay. La modificación debe contemplar que la secretaria general sea la encargada de todos los certificados que brinde el kiosco, lo que permitirá reducir la cantidad necesaria de kioscos a uno por campus, siendo un ahorro significativo para la Universidad.

Al realizar nuevas aplicaciones para consulta de notas, matriculas, horarios, etc., p. ej. apps para dispositivos móviles, se recomienda hacer uso de los servicios web implementados. Con ello, se tiene un middleware común que facilita el mantenimiento de los sistemas de una manera más unificada. Se recomienda, además, extender los servicios web prestados, para que estos puedan ser utilizados por más aplicaciones de distintos propósitos.

Como parte de la mejora en los procesos administrativos de la Universidad del Azuay, dentro de los cuales se encuentran la emisión certificados, se recomienda la realización de un sistema que facilite a los estudiantes la realización de solicitudes mediante el portal web. Al mismo tiempo, este sistema debe permitir el procesamiento ágil de las mismas por parte del personal administrativo, al cual, debe dotársele de la información y herramientas necesarias para la toma de decisiones y el procesamiento automático de lo solicitado. Para este efecto, también se podrá hacer uso de servicios web.

El sistema desarrollado fue pensado para los estudiantes que pertenezcan al sistema de créditos de la Universidad. Sin embargo, se recomienda que también se preste este servicio a los estudiantes de mayor antigüedad. Para lograr esto, solamente es necesario modificar los servicios web, para que diferencien entre los estudiantes de un sistema y del otro, y devuelvan la información correspondiente, con lo que el resto del sistema no se vería afectado. Como recomendación adicional, se sugiere implementar una solución que permita a los estudiantes graduados gozar de este servicio, buscando la manera de realizar el cobro del certificado por adelantado.

En el kiosco siempre debe estar activa la aplicación de certificados académicos, para evitar problemas de seguridad (sección 2.5.1 Puntos Vulnerables). Para ello, se deben tomar medidas al instalar el sistema en el kiosco, de tal manera que,

cuando se llegue a congelar el programa, el programa sea reiniciado automáticamente, y, cuando se reinicie el kiosco, el sistema operativo inicie automáticamente la aplicación. Para cumplir con ello, se puede hacer uso de Active Directory,

Además, se recomienda implementar pistas de auditoría, en donde se indique, entre otras cosas, que usuario emitió un certificado, el cobro que se le realizó y con qué usuario de la base de datos fue ingresada la deuda, para conocer si fue ingresada mediante el kiosco, secretaría u otra instancia.

Finalmente, se sugiere aumentar el número de certificados disponibles para ser emitidos automáticamente. Priorizando aquellos que sean más solicitados en las secretarías de cada facultad.

10. REFERENCIAS

- Alonso, Gustavo, y otros. «Web Services.» Alonso, Gustavo, y otros. Web Services - Concepts, Architectures and Applications. Springer Verlag, 2004. 354.
- Ambler, Scott. UML 2 Package Diagrams. 2010. 08 02 2012 <<http://www.agilemodeling.com/artifacts/packageDiagram.htm>>.
- Boochm, Grady, James Rumbaugh y Ivan Jacobson. The Unified Modeling Language User Guide. Addison-Wesley Professional, 2005.
- Cortizo, José Carlos. "Service Oriented Architecture." n.d. Universidad Europea de Madrid. 20 05 2012 <<http://www.esi.uem.es/jccortizo/temasConcu/soa.pdf>>.
- Mackenzie, S, S Zhang y W Soukoreff. Text Entry Using Soft Keyboards. Behaviour and Information Technology, 1999.
- Meyer-Otte, Ansgar. 27 04 2007. 02 2012 <http://www.viadee.de/fileadmin/pdf/Vortraege/JasperReports_vs_BIRT.pdf>.
- NetBeans. "Introduction to Web Services." n.d. Netbeans. 01 05 2012 <<http://netbeans.org/kb/docs/websvc/intro-ws.html>>.
- Sears, A. "Improving Touchscreen keyboards: Design Issues and Comparasion with other devices." Interacting with Computers. 1991.
- Seguel, Ricargo. "Itescam." 05 2005. Seguridad en Web Services. 03 06 2012 <<http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r52946.PDF>>.
- Sjöberg, Samuel. A touch screen interface for point-of-sale applications in retail stores. Estocolmo, 2005.

- Slawsky, Richard. "Software Security: The importance of Locking Down your self-service Kiosk." 2011. KioskMarketplace. 11 06 2012
<http://global.networldalliance.com/downloads/white_papers/Kioware_SR_Software-Security_To-Launch.pdf>.
- Sparks systems. sparks systems. 2010. 04 08 2012
<http://www.sparxsystems.com/enterprise_architect_user_guide/modeling_languages/sequenceelements.html>.
- TutorialsPoint. "TutorialsPoint." n.d. UML Tutorial. 02 2012
<<http://www.tutorialspoint.com/images/pdf.jpg>>.
- UML diagrams. UML-diagrams. n.d. 02 08 2012 <<http://www.uml-diagrams.org/package-diagrams.html>>.
- W3C. "Extensive Markup Language (XML)." 26 11 2008. W3C. 27 05 2012
<<http://www.w3.org/TR/REC-xml/>>.
- . "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)." 27 04 2007. W3C. 27 05 2012 <<http://www.w3.org/TR/soap12-part1/>>.
- . "SOAP Version 1.2 Part 2: Adjuncts ." 27 04 2007. W3C. 27 05 2012
<<http://www.w3.org/TR/soap12-part2/>>.
- . "Web Services Architecture." 11 02 2004. W3C. 19 05 2012
<<http://www.w3.org/TR/ws-arch/wsa.pdf>>.
- . "Web Services Description Language (WSDL)." 15 03 2001. W3C. 27 05 2012
<<http://www.w3.org/TR/wsdl>>.
- . "XML-infoset." 04 02 2004. W3C. 27 05 2012 <<http://www.w3.org/TR/xml-infoset/>>.

Waloszek, Gerd. "Interaction Design Guide for Touchscreen Applications." 12 2000.

SAP Design Guild. 07 2012

<<http://www.sapdesignguild.org/goodies/TSDesignGL/TSDesignGL.pdf>>.

Wikipedia. "WSDL." 11 12 2007. wikipedia. 27 05 2012

<http://en.wikipedia.org/wiki/Web_Services_Description_Language>.

Wirespring. "WireSpring.com." n.d. The Real World Benefits and Challenges of

Interactive Kiosks. 03 2012

<http://www.wirespring.com/pdf/kiosk_benefits_and_challenges.pdf>.

11. ANEXOS

11.1 ANEXO 1: CERTIFICADO DE MATRICULA

NOMBRE_SECRETARIA, SECRETARIA(O) DE LA FACULTAD DE
FACULTAD DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, *NOMBRES Y APELLIDOS*, estudiante portador de la cedula Nro. *CEDULA*, se encuentra matriculado(a) en la facultad de *FACULTAD*, carrera *CARRERA*, pensum *PENSUM*, para el periodo *PERIODO*, cursando las siguientes materias:

Código	Materia	Paralelo	Créditos	Porcentaje de Asistencia
CODMAT	NOMBMATERIA	LPARAL	NCred	%ASISTENCIA

Es todo cuanto puedo certificar y autorizo a la parte peticionaria dar al presente el uso más conveniente.

Cuenca, *FECHA*.

NOMBRE_SECRETARIA

SECRETARIA(O) DE LA FACULTAD DE *FACULTAD*

11.2 ANEXO 2: CERTIFICADO DE EGRESADO

NOMBRE_SECRETARIA, SECRETARIA DE LA FACULTAD DE
FACULTAD DE LA UNIVERSIDAD DEL AZUAY

C E R T I F I C A:

Que, *NOMBRES Y APELLIDOS*, estudiante portador de la cedula Nro.
CEDULA, luego de haber aprobado:

Créditos Requeridos

Créditos Optativos

Créditos Extras

Exigidos en la carrera de *CARRERA* pensum de estudios *PENSUM* de la
facultad de *FACULTAD*.

Al haber cumplido los requisitos legales y reglamentarios, obtuvo su calidad
de egresado el *FECHA* con la calificación de *CALIFICACION*, equivalente a
EQUIVALENCIA.

Es todo cuanto puedo certificar y autorizo a la parte peticionaria dar al
presente el uso más conveniente.

Cuenca, *FECHA*.

NOMBRE_SECRETARIA

SECRETARIA(O) DE LA FACULTAD DE *FACULTAD*

11.3 ANEXO 3: CERTIFICADO DE NOTAS HISTÓRICAS

NOMBRE_SECRETARIA, SECRETARIA DE LA FACULTAD DE
FACULTAD DE LA UNIVERSIDAD DEL AZUAY

C E R T I F I C A:

Que, *NOMBRES Y APELLIDOS*, estudiante portador de la cedula Nro. CEDULA, aprobó las siguientes asignaturas en la facultad de *FACULTAD* carrera de *CARRERA* pensum *PENSUM*.

Materias	Créditos	Nota	Condición	Comentario
CICLO 1:				
Periodo Mes Año – Mes Año				
<i>Materia_1_primer_ciclo</i>	<i>Créditos</i>	<i>Nota/50</i>	<i>A/R/F</i>	<i>Nro Matricula</i>
<i>Materia_2_primer_ciclo</i>	<i>Créditos</i>	<i>Nota/50</i>	<i>A/R/F</i>	<i>Nro Matricula</i>
.				
.				
.				
CICLO 2:				
Periodo Mes Año – Mes Año				
<i>Materia_1_segundo_ciclo</i>	<i>Créditos</i>	<i>Nota/50</i>	<i>A/R/F</i>	<i>Nro Matricula</i>
<i>Materia_2_segundo_ciclo</i>	<i>Créditos</i>	<i>Nota/50</i>	<i>A/R/F</i>	<i>Nro Matricula</i>
.				
.				
.				

PROMEDIO GENERAL X/50

CRÉDITOS EXTRA

Materia_seminario Créditos Nota/50 A/R/F

Nomenclatura del Estado de la Materia: A: Aprobado, R: Reprobado y F: Reprobado por Faltas.

Las calificaciones se imponen sobre 50 puntos. Para aprobar una materia se requiere de un mínimo de 30 puntos.

El sistema de equivalencias de la Universidad del Azuay es el siguiente:

0 a 29,9	Reprobado
30 a 34,9	Regular
35 a 39,9	Buena
40 a 44,9	Muy Buena
45 a 50	Sobresaliente

Es todo cuanto puedo certificar y autorizo a la parte peticionaria dar al presente el uso más conveniente.

Cuenca, *FECHA*.

NOMBRE_SECRETARIA

SECRETARIA(O) DE LA FACULTAD DE *FACULTAD*

11.4 ANEXO 4: CERTIFICADO CON EL PENSUM

NOMBRE_SECRETARIA, SECRETARIA DE LA FACULTAD DE
FACULTAD DE LA UNIVERSIDAD DEL AZUAY

C E R T I F I C A:

Que, *NOMBRES Y APELLIDOS*, estudiante portador de la cedula CEDULA, se encuentra matriculado en el pensum PENSUM detallado a continuación, perteneciente a la carrera de CARRERA de la facultad de *FACULTAD*.

Código	Materias	Créditos	Prerrequisito	Tipo
PRIMER CICLO:				
<i>Cod_Materia</i>	<i>Materia_1_primer_ciclo</i>	<i>Créditos</i>	<i>¿??????????</i>	<i>Requerida</i>
<i>Cod_Materia</i>	<i>Materia_2_primer_ciclo</i>	<i>Créditos</i>	<i>¿??????????</i>	<i>Requerida</i>

.

.

.

SEGUNDO CICLO:

<i>Cod_Materia</i>	<i>Materia_1_segundo_ciclo</i>	<i>Créditos</i>	<i>¿??????????</i>	<i>Requerida</i>
<i>Cod_Materia</i>	<i>Materia_2_segundo_ciclo</i>	<i>Créditos</i>	<i>¿??????????</i>	<i>Requerida</i>

.

.

.

Es todo cuanto puedo certificar y autorizo a la parte peticionaria dar al presente el uso más conveniente.

Cuenca, *FECHA*.

NOMBRE_SECRETARIA

SECRETARIA(O) DE LA FACULTAD DE *FACULTAD*

11.5 ANEXO 5: CERTIFICADO CON EL HORARIO

NOMBRE_SECRETARIA, SECRETARIA DE LA FACULTAD DE
FACULTAD DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, *NOMBRES Y APELLIDOS*, estudiante portador de la cedula Nro. *CEDULA*, se encuentra matriculado/a en la facultad de *FACULTAD* carrera de *CARRERA* pensum *PENSUM*, para el periodo *PERIODO*, teniendo asignado el siguiente horario:

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO
HH:MM – HH_F:MM_F	MATERIA	MATERIA	MATERIA	MATERIA	MATERIA	MATERIA
.						
HH:MM – HH_F:MM_F	MATERIA	MATERIA	MATERIA	MATERIA	MATERIA	MATERIA

Es todo cuanto puedo certificar y autorizo a la parte peticionaria dar al presente el uso más conveniente.

Cuenca, *FECHA*.

NOMBRE_SECRETARIA

SECRETARIA(O) DE LA FACULTAD DE *FACULTAD*

11.6 ANEXO 6: CERTIFICADO DE NOTAS DEL CICLO ACTUAL

NOMBRE_SECRETARIA, SECRETARIA DE LA FACULTAD DE
FACULTAD DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, *NOMBRES Y APELLIDOS*, estudiante portador de la cedula Nro. *CEDULA*, se encuentra matriculado/a en la facultad de *FACULTAD* de la carrera de *CARRERA* y ha obtenido las siguientes calificaciones en el periodo *PERIODO*:

Código	Materia	Nivel	Créditos	Aporte	Examen	Susp	Final
<i>Cod_Materia</i>	<i>Materia_1</i>	<i>NN</i>	<i>Ncred</i>	<i>X/30</i>	<i>X/20</i>	<i>X/20</i>	<i>X/50</i>
<i>Cod_Materia</i>	<i>Materia_2</i>	<i>NN</i>	<i>Ncred</i>	<i>X/30</i>	<i>X/20</i>	<i>X/20</i>	<i>X/50</i>
<i>Cod_Materia</i>	<i>Materia_3</i>	<i>NN</i>	<i>Ncred</i>	<i>X/30</i>	<i>X/20</i>	<i>X/20</i>	<i>X/50</i>
<i>Cod_Materia</i>	<i>Materia_4</i>	<i>NN</i>	<i>Ncred</i>	<i>X/30</i>	<i>X/20</i>	<i>X/20</i>	<i>X/50</i>

PROMEDIO PERIODO *X/50*

Las calificaciones se imponen sobre 50 puntos. Para aprobar una materia se requiere de un mínimo de 30 puntos.

Es todo cuanto puedo certificar y autorizo a la parte peticionaria dar al presente el uso más conveniente.

Cuenca, *FECHA*.

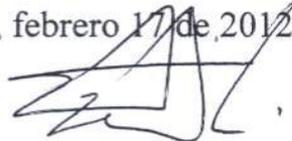
NOMBRE_SECRETARIA

SECRETARIA(O) DE LA FACULTAD DE *FACULTAD*

**DOCTOR ROMEL MACHADO CLAVIJO,
SECRETARIO DE LA FACULTAD DE CIENCIAS DE LA
ADMINISTRACION
DE LA UNIVERSIDAD DEL AZUAY,
CERTIFICA:**

Que, el H. Consejo de Facultad se sesión realizada el 10 de febrero de 2013, conoció la petición presentada por los señores **Alfonso José Rivera Donoso con código 44310 y Diego Cedillo Durán con código 43428**, que denuncian su tema de tesis denominado: **“SISTEMA INFORMÁTICO PARA KIOSCOS EMISORES DE CERTIFICADOS ACADÉMICOS PARA ESTUDIANTES DE LA UNIVERSIDAD DEL AZUAY”**, presentado como requisito previo a la obtención del Grado de Ingeniero de Sistemas. El Consejo acoge el informe de la Junta Académica y lo aprueba. Designa como Directora a la ing. Janela Encala y como miembros del Tribunal Examinador a los ings. Marcos Orellana y Lenin Erazo. De conformidad con las disposiciones reglamentarias los denunciados deberán presentar su trabajo de tesis en un lapso no mayor a **DIECIOCHO MESES**, contados a partir de la fecha de aprobación, esto es hasta el 10 de agosto de 2013.-

Cuenca, febrero 17 de 2013





Universidad del Azuay

Facultad de Ciencias de la Administración
Escuela de Ingeniería de Sistemas

Sistema informático para kioscos emisores de certificados académicos para estudiantes de la Universidad del Azuay

Diseño de Tesis

Autores:

Diego Xavier Cedillo Durán

Alfonso José Rivera Donoso

Director propuesto:

Ing. Janela Encalada

Cuenca - Ecuador

2011

Sistema informático para kioscos emisores de certificados académicos para estudiantes de la Universidad del Azuay

Antecedentes

En la Universidad del Azuay se ha podido observar que existen demoras injustificadas en el momento de otorgar certificados a los estudiantes, teniendo que esperar un día para obtener el certificado pedido. El proceso del sistema actual comienza cuando un estudiante se acerca a la solicitar un certificado, las secretarias de la Facultad lo redactan y se espera hasta que el secretario lo firme, siendo estas dos últimas fases las causantes de la demora; problema que podría ser resuelto con un sistema que automatice el procedimiento descrito en el *Manual de Procesos de Certificaciones Estándar*, el cual es el resultado de una consultoría realizada a la Universidad del Azuay por “Dynamix Solutions Consultores Integrales”.

Selección y delimitación del tema

El tema comprende:

1. Análisis, Diseño, y elaboración de un sistema interactivo para la generación de certificados académicos, dejando implementados los descritos en el *Manual de Procesos de Certificaciones Estándar*, que son:

- a. Certificado de Notas.
- b. Certificado del horario de clases.
- c. Certificado del *pensum* de la carrera.
- d. Certificado de matrícula.

(Dynamix Solutions Consultores Integrales)

Adicionalmente, se implementará el Certificado de egresado.



2. Análisis, Diseño y elaboración de procedimientos almacenados o *web services* para generar cargos, identificar al usuario, y otorgar los datos necesarios para los certificados descritos.
3. Cobro automático por los certificados entregados, por medio de la generación de la deuda en la cuenta del estudiante; razón por la cual, el sistema estará enfocado para estudiantes actualmente matriculados.
4. Para el funcionamiento del sistema se supone que el kiosco utilizará hojas de papel universitario, con las debidas seguridades para evitar su falsificación; para lo cual, luego de un estudio realizado, recomendamos colocar en estas hojas un sello seco y, posteriormente, firmarlas y sellarlas debidamente por la secretaría general de la Universidad, tal como lo permite el reglamento general.

Justificación

Los estudiantes son parte importante de la Universidad, y por tanto, se les debe brindar el mejor servicio en el menor tiempo posible. En la actualidad, para el caso de emisión de certificados, esta premisa no se cumple, pues se ha observado que se requiere múltiples visitas al campus para finalmente obtener el certificado. Por esto resulta necesario implementar un punto de atención automático para emitir los certificados que los alumnos requieran, brindándoles una atención inmediata.

Descripción del objeto de estudio

El sistema podrá ser implementado en un kiosco impresor, y permitirá receptor las peticiones de certificados, recargar su costo a la matrícula y entregar los certificados inmediatamente, sin que pierdan su validez y legalidad, brindando las mayores facilidades a los alumnos de la Universidad.

Objetivo general

Desarrollar un sistema informático para kioscos emisores de certificados académicos para estudiantes de la Universidad del Azuay.

Objetivos específicos

- Desarrollar un sistema para interfaz táctil, que permita emitir certificados académicos.
- Desarrollar los procedimientos almacenados necesarios para que sean utilizados por el sistema propuesto.
- Validar el software desarrollado para brindar el servicio propuesto.

Metodología

La investigación tendrá un enfoque cuali-cuantitativo. Se utilizará el método de análisis orientado a objetos para el modelado del problema y de su posterior solución. Se obtendrá información a partir de la consultoría realizada en la Universidad del Azuay, en particular lo correspondiente al *Manual de Procesos de Certificaciones Estándar*; y entrevistas al personal administrativo de la Universidad. Con esta información se elaborará un diagrama de casos de uso que represente el comportamiento entre el usuario y el kiosco; y los diagramas que resulten necesarios.



El desarrollo de la aplicación se lo realizará en el lenguaje de programación orientada a objetos JAVA.

Marco teórico

Kiosco

Un kiosco interactivo es un computador especialmente diseñado para, dentro de un acceso público, proveer un autoservicio (*self-service*) de información de distintas índoles de acuerdo al lugar en donde se encuentre. Estos kioscos se los utiliza en lugares totalmente públicos, como plazas, calles, centros comerciales, aeropuertos: para proveer información de lugares turísticos, hoteles, restaurantes, centros de entretenimiento. Pero también se los encuentra en lugares más privados, como bancos, empresas, universidades, etc. entregando información de la propia institución.

(Custom Kiosk)

Estos puntos informativos se componen de hardware seleccionado específicamente para cumplir las necesidades de cada institución. En la mayoría de los casos, está compuesto por una pantalla táctil (*touch*) que contiene dentro de su mismo armazón, el procesador, la memoria, el disco duro y la tarjeta madre. Dependiendo de la cantidad de información que debe tomar del usuario, puede poseer un teclado y un ratón. Una cámara de vídeo, parlantes y un micrófono son necesarios si el kiosco permite la oportunidad de comunicarse con una operadora. Y si su misión no es únicamente informativa, puede poseer una impresora que le permita otorgar reportes.

turnos, certificados, etc. Otros componentes extras pueden servir de ayuda para entregar mayores facilidades al usuario.

El software utilizado puede ser simple o complejo de acuerdo al servicio que vaya a prestar. Si el kiosco es meramente informativo, el software será el más simple posible, con mayor cantidad de imágenes que permitan llamar la atención del informado. A medida que existe mayor necesidad de interacción con el usuario, se necesita mayor complejidad en el software a desarrollar, llegando a hacer uso de complejas consultas y procedimientos que permitan imprimir reportes con la información solicitada. (Kiosk Market place)

Certificados

Un certificado es un tipo de texto administrativo empleado para constatar un hecho determinado. En el proceso de solicitud de un trabajo o de una admisión en una universidad, los certificados son fundamentales para demostrar la formación y experiencia de la persona que presenta sus papeles. El certificado se produce normalmente a instancias de quien lo recibe, y es emitido por una persona con autoridad suficiente dentro de la institución, la cual debe establecer que se ha cumplido con lo afirmado en el documento. (Certificados)

Análisis y Diseño de Sistemas Orientados a Objetos

“El análisis y diseño orientado a objetos es un enfoque cuyo propósito es facilitar el desarrollo de sistemas que deben cambiar con rapidez en respuesta a entornos de negocios dinámicos.” (Kenneth E. Kendall)

“El análisis orientado a objetos y el diseño orientado a objetos se basan en los conceptos de la programación orientada a objetos.” Estos han sido estandarizados en UML, un lenguaje de modelado de datos que permite mejorar la calidad del análisis y diseño del sistema, contribuyendo a la construcción de sistemas de calidad. Entre los



diagramas UML se encuentran el diagrama de casos de uso, de actividad, de secuencia, de clases entre otros. (Kenneth E. Kendall)

La programación orientada a objetos se diferencia de la programación tradicional de procedimientos en que estudia los objetos que forman parte del sistema.

Un Objeto es la representación informática de una entidad real o abstracta. Estos objetos son agrupados en clases, para facilitar la reutilización y mantenimiento. En cada clase se define los atributos y métodos comunes de los objetos que la conforman.

(Kenneth E. Kendall)

Esquema tentativo

Capítulo I: Levantamiento de la Información Sistema Táctil emisor de certificados

Estudio de la consultoría realizada.

Entrevista al personal administrativo.

Análisis Orientado a Objetos del problema (AOO).

Revisión y análisis de la estructura y funcionalidad de las bases de datos de relacionadas con estudiantes y *pénsums* académicos.

Diseño de la arquitectura del sistema.

Capítulo II: Implementación de *web services*

Diseño de procesos.

Diseño de interfaces externas.

Desarrollo de los procedimientos almacenados.

Documentación de procedimientos

Pruebas de procedimientos.

Capítulo III: Implementación del Sistema de Certificados

Diseño de Interfaces Visuales.

Diseño de Procesos.

Desarrollo de los certificados académicos.

Pruebas de funcionamiento.

Pruebas con datos reales.

Conclusiones

Recomendaciones

Bibliografía

Bibliografía

- "Arc Design." n.d. 11 12 2011 <<http://www.arcdesignconsulting.com/custom-kiosk.php>>.
- Dynamix Solutions Consultores Integrales. Diagrama de Flujo. Certificaciones estándar. Resultado de una Consultoría. Cuenca, 2011.
- . Manual de Procesos. Certificaciones Estándar. Resultado de una Consultoría. Cuenca, 2011.
- Kenneth E. Kendall, Julie E. Kendall. Análisis y diseño de sistemas. México: PRENTICE HALL, 2005.
- "Kiosk Market place." n.d. Software Security. 11 12 2011 <http://global.networldalliance.com/downloads/white_papers/Kioware_SR_Software-Security_To-Launch.pdf>.
- "Wikipedia." n.d. Certificado. 04 08 2011 <<http://es.wikipedia.org/wiki/Certificado>>.