



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**IMPLEMENTACIÓN DEL SISTEMA DE GENERACIÓN DE INFORMES, EDICIÓN DE
DOCUMENTOS Y DIGITALIZACIÓN DE CONTROL DE CALIDAD EN LA FACULTAD DE
FARMACIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Edwin Alexander Sánchez León

Asesorado por la Inga. Gladys Sucely Aceituno

Guatemala, junio de 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

IMPLEMENTACIÓN DEL SISTEMA DE GENERACIÓN DE INFORMES, EDICIÓN DE DOCUMENTOS Y DIGITALIZACIÓN DE CONTROL DE CALIDAD EN LA FACULTAD DE FARMACIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

EDWIN ALEXANDER SÁNCHEZ LEÓN
ASESORADO POR LA INGA. GLADYS SUCELY ACEITUNO

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, JUNIO DE 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Walter Rafael Véliz Muñoz
VOCAL V	Br. Sergio Alejandro Donis Soto
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADORA	Inga. Sonia Yolanda Castañeda Ramírez
EXAMINADORA	Inga. Floriza Ávila Pesquera de Medinilla
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

IMPLEMENTACIÓN DEL SISTEMA DE GENERACIÓN DE INFORMES, EDICIÓN DE DOCUMENTOS Y DIGITALIZACIÓN DE CONTROL DE CALIDAD EN LA FACULTAD DE FARMACIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 30 de abril de 2013.


Edwin Alexander Sánchez León

Guatemala, 17 de mayo de 2014

Ingeniero Silvio José Rodríguez Serrano
Director de la Unidad de EPS
Facultad de Ingeniería
Presente

Estimado Ingeniero Rodríguez Serrano.


Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado (E.P.S.) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Edwin Alexander Sánchez León**, carné No. **200117120** procedí a revisar el informe final, cuyo título es **"Implementación de Sistema de Generación de Informes, Edición de Documentos y Digitalización de Control de Calidad en la Facultad de Farmacia de la Universidad de San Carlos de Guatemala"**.

En tal virtud, **LO DOY POR APROBADO**, solicitándole el trámite respectivo.

Sin otro particular, me es grato suscribirme,

Atentamente,

"Id y Enseñad a Todos"


Inga. Gladys Aceituno
Asesora de EPS





Guatemala, 26 de mayo de 2014.
REF.EPS.DOC.615.05.2014.

Ing. Silvio José Rodríguez Serrano
Director Unidad de EPS
Facultad de Ingeniería
Presente

Estimado Ingeniero Rodríguez Serrano .


Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Edwin Alexander Sánchez León** carné No. **200117120** procedí a revisar el informe final, cuyo título es **IMPLEMENTACIÓN DE SISTEMA DE GENERACIÓN DE INFORMES, EDICIÓN DE DOCUMENTOS Y DIGITALIZACIÓN DE CONTROL DE CALIDAD EN LA FACULTAD DE FARMACIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA.**

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

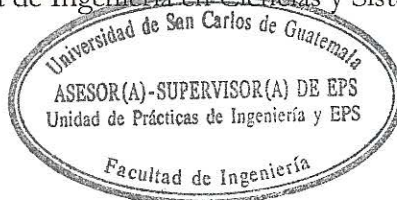
Sin otro particular, me es grato suscribirme.

Atentamente,

"Id y Enseñad a Todos"


Inga. Floriza Felipa Avila Pesquera de Medinilla
Supervisora de EPS

Área de Ingeniería en Ciencias y Sistemas



FFAPdM/RA



Guatemala, 26 de mayo de 2014.
REF.EPS.D.298.05.2014.

Ing. Marlon Antonio Pérez Turk
Director Escuela de Ingeniería Ciencias y Sistemas
Facultad de Ingeniería
Presente

Estimado Ingeniero Perez Turk.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **IMPLEMENTACIÓN DE SISTEMA DE GENERACIÓN DE INFORMES, EDICIÓN DE DOCUMENTOS Y DIGITALIZACIÓN DE CONTROL DE CALIDAD EN LA FACULTAD DE FARMACIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, que fue desarrollado por el estudiante universitario **Edwin Alexander Sánchez León carné No. 200117120** quien fue debidamente asesorado por la Inga. Sucely Aceituno y supervisado por la Inga. Floriza Felipa Ávila Pesquera de Medinilla.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte de la Asesor y la Supervisora de EPS, en mi calidad de Director apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,
"Id y Enseñad a Todos"

Ing. Silvio José Rodríguez Serrano
Director Unidad de EPS



SJRS/ra



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 4 de Junio de 2014


Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación-EPS del estudiante **EDWIN ALEXANDER SÁNCHEZ LEÓN**, carné **2001-17120**, titulado: **"IMPLEMENTACIÓN DE SISTEMA DE GENERACIÓN DE INFORMES, EDICIÓN DE DOCUMENTOS Y DIGITALIZACIÓN DE CONTROL DE CALIDAD EN LA FACULTAD DE FARMACIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



E
S
C
U
E
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“IMPLEMENTACIÓN DEL SISTEMA DE GENERACIÓN DE INFORMES, EDICIÓN DE DOCUMENTOS Y DIGITALIZACIÓN DE CONTROL DE CALIDAD EN LA FACULTAD DE FARMACIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, realizado por el estudiante EDWIN ALEXANDER SÁNCHEZ LEÓN, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

Ing. *Marion Antonio Pérez Türk*
Director, Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 23 de junio 2014



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **IMPLEMENTACIÓN DEL SISTEMA DE GENERACIÓN DE INFORMES, EDICIÓN DE DOCUMENTOS Y DIGITALIZACIÓN DE CONTROL DE CALIDAD EN LA FACULTAD DE FARMACIA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario: **Edwin Alexander Sánchez León** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.


Ing. Murphy Olympo Paiz Resinos
Decano



Guatemala, junio de 2014

ACTO QUE DEDICO A:

- Dios** Por darme la oportunidad de culminar mis estudios universitarios al proveerme de salud, medios materiales, guía espiritual, salud física y mental y capacidades para ello, por ayudarme a confiar en mí a pesar de todo, por el aire que respiro, por mi cuerpo, por vivir en Guatemala, por tener una familia; por eso y porque sin Él nada de esto sería posible.
- Mis padres** Por su apoyo, consejos e invaluable ayuda en todo momento y ser una fuente de inspiración para superarme.
- Universidad de San Carlos de Guatemala** Porque gracias a su servicio de enseñanza obtuve el conocimiento para desarrollarme como profesional.
- Mis tíos** Gonzalo León, Violeta León, Danilo Guzmán, Reina Herrera, Mildred León, por los consejos y por instarme a seguir y perseverar.

AGRADECIMIENTOS A:

**Universidad de
San Carlos de
Guatemala**

Gracias a su servicio de enseñanza obtuve el conocimiento para desarrollarme como profesional.

Mis tíos

Gonzalo León, Violeta León, Danilo Guzmán, Reina Herrera, Mildred León. Por los consejos y por instarme a seguir y perseverar.

Mis hermanos

Gatsby y Alan Sánchez, por su compañía y su amor y hacerme ver la realidad.

Mis primos

Carol, Andrea, Juan Pablo, Marielos, Celeste Hernández, por infundirme entusiasmo y motivación con su compañía y buen humor.

Mis terapeutas

Myriam López y María del Rosario Sim, por ayudar a encontrarme a mí mismo a trascender y avanzar en el laborioso camino de convertir en luz mi oscuridad.

Mis amigos

Ricardo Cruz Ortíz, Rodolfo Zea y Wendy Rodríguez de Ramos, quienes siempre me instaron a creer en mí mismo y superarme para alcanzar mis objetivos.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	VII
RESUMEN	XVII
OBJETIVOS	XIX
INTRODUCCIÓN	XXI
1. ANTECEDENTES	1
1.1. Descripción de la lógica de negocio	2
1.2. Aspecto financiero	2
1.3. El proceso de inmuno-diagnóstico	3
1.4. Análisis de la muestra.....	3
1.5. Elaboración de diagnósticos	5
2. MARCO TEÓRICO.....	7
2.1. Consideraciones técnicas.....	9
2.1.1. Nivel de visibilidad	9
2.1.2. Software.....	9
2.1.3. Sistema operativo anfitrión	11
2.1.4. Dependencias.....	11
2.1.5. Hardware	12
2.1.6. Factores de calidad.....	12
2.2. Consideraciones técnicas del desarrollo del software	14
2.2.1. Diseño de módulos (diagrama Entidad-Relación)....	14

3.	PRELIMINARES	17
3.1.	Entrevistas	17
3.2.	Toma de requerimientos	18
3.3.	Negociaciones de la aplicación	19
3.4.	Casos de uso	19
4.	ANÁLISIS DE LA SOLUCIÓN	21
4.1.	Consideraciones técnicas	21
4.1.1.	.NET	21
4.1.2.	SQL Server	22
4.2.	Explicación de la arquitectura de software	23
4.2.1.	Aplicación de escritorio	23
4.3.	¿Por qué usar Extreme Programming (XP)?	23
4.3.1.	Pros y contras de la metodología	24
4.3.2.	Forma de implementación	25
5.	DESARROLLO DE LOS MÓDULOS	29
5.1.	Login	29
5.2.	Ingreso de muestras	30
5.3.	Generar informe	32
5.4.	Administrar informes	33
5.5.	Estadísticas	34
5.6.	Seguridad	35
5.7.	Documentos	36
5.8.	Desarrollo del módulo de <i>checklist</i>	37
5.9.	Desarrollo del módulo de bitácora	39
5.10.	Desarrollo del módulo de inventario	40

6.	CAPACITACIÓN.....	41
6.1.	Plan de trabajo.....	41
6.1.1.	Planificación.....	41
6.1.2.	Realización de la capacitación.....	41
6.2.	Finalización de la capacitación y entrega de los productos finales.....	42
6.3.	Beneficios de la aplicación.....	42
	CONCLUSIONES	45
	RECOMENDACIONES	47
	BIBLIOGRAFÍA	49
	APÉNDICES	51

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Diagrama Entidad-Relación de los módulos	15
2.	Casos de Uso de cada uno de los módulos	20
3.	Diagrama de flujo	29
4.	Captura en pantalla de la versión definitiva del módulo Login en tiempo de ejecución	30
5.	Diagrama de flujo para el módulo de ingreso de muestras	30
6.	Versión final del módulo de ingreso de muestras.....	31
7.	Diagrama de flujo del módulo de generación de informes	32
8.	Versión final del módulo para generar informe.....	33
9.	Diagrama de flujo del módulo de modificación de informe	33
10.	Versión final del módulo para administrar informes.....	34
11.	Versión final del módulo de estadísticas	35
12.	Diagrama de flujo del módulo de edición de documentos	36
13.	Versión final del módulo documentos.....	37
14.	Diagrama de flujo del módulo <i>checklist</i>	38
15.	Versión final del módulo <i>checklist</i>	38
16.	Diagrama de flujo del módulo bitácora	39
17.	Diagrama de flujo del módulo inventario	40

GLOSARIO

AJAX	Acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).
ASP	Active Server Pages, también conocido como ASP clásico, es una tecnología de Microsoft del tipo lado del servidor para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS).
ASP.NET	Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.
Bases de datos	Es una recopilación de información relativa a un asunto o propósito particular, como el seguimiento de pedidos de clientes o el mantenimiento de una colección de música.
<i>Bussiness Intelligence</i>	Inteligencia -Empresarial. Sus siglas en inglés son BI.

<i>Bussiness Intelligence Development Studio</i>	Es un componente de Microsoft Visual Studio 2008 con tipos de proyecto adicionales que son específicos de Business Intelligence de SQL Server.
<i>CheckList</i>	Término del inglés que suele traducirse como lista de control, lista de comprobación o lista de verificación.
<i>Client Side</i>	Aplicación vista desde el lado del cliente.
CLR	Denominado Common Language Runtime o CLR (entorno en tiempo de ejecución de lenguaje común) es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma.
Coliformes	Denominación genérica designa para nombrar a un grupo de especies bacterianas que tienen ciertas características bioquímicas en común e importancia relevante como indicadores de contaminación del agua y los alimentos.
DataBind	Microsoft .NET que también es el encargado de compilar una forma de código intermedio Intermediate Language (CIL, anteriormente conocido como MSIL, por Microsoft Intermediate Language).

DataSet	Función propia de los controles de tablas de <i>Visual Studio</i> para poder cargar y refrescar datos en tiempo de ejecución.
Digidyne	Objeto para controlar y manipular bases de datos desde <i>Visual Studio</i> sin interactuar directamente con la Base de datos y sin escribir sentencias SQL.
Document Object Model	Es un programa de consultoría tecnológica que ofrece servicios firmes a empresas pequeñas, medianas y grandes en el área metropolitana de Nueva York. (Modelo en objetos). Es una interfaz de programación de aplicaciones (API) para representar documentos (En español 'Modelo de Objetos del Documento' o HTML y XML en un modelo para combinar dichos objetos y manipularlos.
DOM	Siglas de Document Object Model.
E. Coli	Vèase Esterichia Coli.
EDI	Llamado también IDE (sigla en inglés de <i>integrated development environment</i> , Entorno de Desarrollo Integrado), es un programa informático compuesto por un conjunto de herramientas de programación.

EMAScript	Es una especificación de lenguaje de programación publicada por ECMA International actualmente está aceptado como el estándar ISO 16262.
Escherichia coli	Es una enterobacteria que se encuentra generalmente en los intestinos animales, y las aguas negras.
Firefox	Término usado comúnmente para referirse de forma abreviada al navegador Mozilla Firefox.
Framework	Un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.
Grid	Control utilizado en Visual Studio y ASP.NET para mostrar tablas en las páginas o ventanas de la aplicación. Soporta conexión a Bases de dato y sus interacciones con las mismas.
HTTP	Siglas en inglés de HyperText Transfer Protocol (en español Protocolo de Transferencia de Hipertexto).

IBM	<i>International Business Machines (IBM)</i> . Es una empresa multinacional estadounidense de tecnología y consultoría que fabrica y comercializa hardware y software para computadoras, ofreciendo servicios de infraestructura, alojamiento de Internet y otras áreas de la informática desde computadoras centrales hasta nanotecnología.
IDE	Siglas en inglés de Entorno de Desarrollo Integrado (<i>integrated development environ-ment</i>).
IIS	Siglas de Internet Information Services. También llamado Internet Information Services es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows.
Inmuno – Diagnóstico	Son técnicas, basadas en la reacción de antígeno, anticuerpo, que se aplican en el estudio del proceso infeccioso.
JavaScript	JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMA Script.
LAMIR	Siglas del Laboratorio Microbiológico de Referencia.

<i>MerdiSystem</i>	Sistema de Salud que ofrece servicios para los estadounidenses con componentes básicos para crear soluciones a nivel empresarial.
<i>Mozilla</i>	Término usado comúnmente para referirse de forma abreviada al navegador Mozilla Firefox.
<i>Mozilla Application Suite</i>	Es un navegador web y una plataforma de desarrollo libre y de código abierto. Por decisión de la Fundación Mozilla esta suite ha dejado de ser desarrollada siendo actualmente su sucesor <i>SeaMonkey</i> .
<i>Mozilla Firefox</i>	Es un navegador web libre y de código abierto descendiente de <i>Mozilla Application Suite</i> y desarrollado por la Fundación <i>Mozilla</i> .
<i>Obout Suite</i>	Es una serie de librerías que contienen controles para aplicaciones de ASP.NET.
OEM	Siglas de <i>Original Equipment Manufacturer</i> (en español de Fabricante de Equipo Original).
RIA	Del rich Internet <i>applications</i> , o RIA (en español aplicaciones de Internet enriquecidas), son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales.

***Scripting Language
Server Side***

Es un lenguaje de programación que soporta la escritura de scripts, programas escritos para entornos de software para automatizar la ejecución de tareas.

Service Pack

Término para describir una vista de una aplicación desde el lado del Servidor. Los programas denominados como *Service Pack* (o en la sigla en inglés SP) consisten en un grupo de parches que actualizan, corrigen y mejoran aplicaciones y sistemas operativos.

Shigella

Esta denominación fue popularizada por Microsoft cuando comenzó a empaquetar grupos de parches que actualizaban su sistema operativo Windows.

ShcLab

Es un género de bacterias en forma de bacilo, no formadora de esporas e incapaces de fermentar la lactosa, que pueden ocasionar diarrea en los seres humanos.

Sourcing

Empresa dedicada al desarrollo de software para laboratorios de análisis clínicos.

SQIServer

En términos de negocios, se refiere a una serie de prácticas de contratación, con el fin de encontrar, evaluar y contratación de proveedores de bienes y servicios.

Staphylococcus aureus

SQL Server 2005 es una plataforma global de base de datos que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial.

TIMLAB

Bacteria anaerobia facultativa, grampositiva, productora de coagulasa, catalasa, inmóvil y no esporulada que se encuentra ampliamente distribuida por todo el mundo, estimándose que una de cada tres personas se hallan colonizadas, que no infectadas, por ella.

Tipado dinámico

Software para laboratorio que permite la administración de las operaciones habituales en un laboratorio y permite realizar estadísticas y reportes para medir la productividad y efectividad del laboratorio.

Visual Basic .NET

Un lenguaje de programación es dinámicamente tipado si una misma variable puede tomar valores de distinto tipo en distintos momentos.

Visual C#

Es un lenguaje de programación orientado a objetos que se puede considerar una evolución de *Visual Basic* implementada sobre el framework .NET.

Visual C++	También conocido como MSVC, (Microsoft Visual C++). Es un entorno de desarrollo integrado (IDE) para lenguajes de programación C, C++ y C++/CLI.
Visual J#	Es un lenguaje transicional para programadores del de programación Java y del lenguaje J++ de Microsoft, creado con la intención de que ambos pueden usar sus conocimientos actuales para crear aplicaciones en la plataforma .NET de Microsoft.
Visual Studio	Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET.
Widget	Es una pequeña aplicación o programa, usualmente presentada en archivos o ficheros pequeños que tiene como objetivo dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.
Wizard	Miniprograma asistente que permite preparar, configurar y colocar objetos, facilitando la programación y el uso de los mismos.

Windows 7 UE

También denominada *Ultimate Edition*. Esta edición es igual a la versión Enterprise pero sin las restricciones de licenciamiento por volumen, permitiéndose su compra en canales de venta al público general, aunque Microsoft ha declarado que en lugar de publicitarse

XMLHttpRequest

Es una clase de ASP.NET que permite leer los valores HTTP enviados por un cliente durante una solicitud Web.

RESUMEN

El Laboratorio Microbiológico de Referencia, LAMIR, en su área de microbiología, es una entidad encargada de elaborar análisis a partir de muestras, ingresando previamente estas y registrando dicha operación y posteriormente elaborando el análisis, el cual puede ser modificado e impreso, además de llevar el control de calidad.

También en el LAMIR se hace registro de inventario de las herramientas e insumos con los que cuenta el laboratorio para sus operaciones, además de tener una serie de documentos y manuales con información y contenido afín a dichas operaciones.

Para llevar a cabo la acreditación del laboratorio, se elaboró una serie de cambios y reformas entre las que están: elaboración de un programa que digitalice las operaciones que el LAMIR realiza actualmente, por lo cual el programa deberá llevar registro del ingreso de las personas encargadas, ingreso de muestras tipo de las mismas, elaboración de análisis e informes con base en las mismas muestras, impresión y edición de datos ingresados, elaboración de documentos y edición de manuales, registro de inventario y control de calidad con base en formularios para ello.

OBJETIVOS

General

Desarrollar un software referente de los servicios de microbiología del LAMIR y que responda a los requerimientos del mismo, así también esté alineado a los objetivos de dicha entidad.

Específicos

1. Crear módulos que permita registrar informes y muestras así como almacenarlas e imprimirlas.
2. Desarrollar módulos para edición y muestra de documentos.
3. Desarrollar un módulo de control para crear *checklist* para control de calidad.
4. Desarrollar un módulo de bitácora de eventos.
5. Desarrollar un módulo de generación de estadísticas.
6. Desarrollar un módulo de registro de equipo.
7. Desarrollar un módulo de bitácora de eventos.

INTRODUCCIÓN

El presente trabajo es un informe de actividades, así como los esquemas a nivel técnico y conceptual en los cuales se han basado para la elaboración del proyecto.

Es en estos momentos cuando la automatización informática tiene auge alrededor del mundo, tanto por las facilidades que otorga el apoyo en la tecnología para ciertas operaciones mecánicas o cuyas operaciones se pueden delegar a los ordenadores, como la necesidad de estar al corriente de los avances tecnológicos que están revolucionando la forma en cómo la humanidad procesa, crea y accede a la información.

Por tal motivo, se han desarrollado tendencias como el desarrollo orientado a servicios, donde se hace énfasis a la generación de aplicaciones cumpliendo ciertos estándares para una industria o trabajo específico; no obstante, no es nada fuera de lo común encontrarse con aplicaciones hechas a la medida, para satisfacer específicamente una necesidad de cierta forma que una aplicación más genérica no podría hacer.

Es precisamente por un motivo de especificidad y detalle altos que surgió la necesidad de un proyecto de esta naturaleza para el LAMIR y el cual es elaborado con el objetivo de ser un apoyo y una herramienta para el beneficio del LAMIR y así contribuir a sus objetivos e intereses como entidad al servicio de la sociedad guatemalteca.

En las siguientes páginas se encontrarán detallados todos los pasos, tanto en actividades como esquemas lógicos, consideraciones técnicas y elaboraciones a nivel software y hardware, así como del desarrollo, herramientas utilizadas soluciones propuestas y esquemas de elaboración, bitácora de actividades, entre otras, todos los pasos y acciones efectuadas para este proyecto, con el cual se espera sea un comienzo en el uso de la tecnología informática para el LAMIR y el inicio en la búsqueda de innovaciones y modificaciones para el mejoramiento de la entidad.

1. ANTECEDENTES

A nivel global, hay software orientado a laboratorios tales como TIMLAB, MediSystem o SlcLab, que son herramientas de software con el fin de ofrecer soluciones y servicios para empresas en la rama de los laboratorios microbiológicos y clínicos.

A nivel más global, si bien no son aplicaciones especializadas, se encuentran en entidades como DIAGNOSON, y otro tipo de laboratorios nacionales donde facturan y entregan informes o reportes que utilizan software, en algunos casos rudimentario, sin llegar a una evolución de software con entornos gráficos o ambientes amigables, sino usando entornos más rudimentarios para ambientes Windows, llegando incluso al clásico modo texto al estilo DOS.

El Laboratorio Microbiológico de Referencia (LAMIR) es una entidad que pertenece a la escuela Química Biológica de la Facultad de Ciencias Químicas y Farmacia. La misma empezó funciones desde 1994.

Con laboratorios como el LAMIR se pretende mejorar la calidad de vida de los guatemaltecos, formando profesionales capacitados en el campo del inmuno diagnóstico y la microbiología y las aplicaciones de los mismos, prestando servicios de alta calidad y la generando información útil para la toma de decisiones analizando resultados y elaborando conclusiones de los mismos, así como de nuevos conocimientos a través de la investigación.

El LAMIR extiende sus servicios a los sectores de salud, medio ambiente, agrícola e industrial y busca ser reconocido a nivel nacional e internacional, acreditado y que cuente con personal con compromiso y responsabilidad, siendo además altamente calificado.

A continuación se describe al laboratorio desde el punto de vista como negocio y por los servicios que ofrece.

1.1. Descripción de la lógica de negocio

Laboratorio Microbiológico de Referencia (LAMIR) es una entidad que se dedica a ofrecer servicios de análisis microbiológico y el inmuno - diagnóstico a entidades privadas y públicas, además de empresas particulares. Actualmente tienen un sistema de control de inventarios basados en registros de tipo manual.

1.2. Aspecto financiero

El LAMIR cobra por los servicios que ofrece para tener los recursos económicos que le permitan proveerse del equipo e insumos necesarios para desempeñar sus actividades y análisis.

Actualmente el LAMIR es una entidad conocida y con prestigio en el medio, por lo que forma parte de los laboratorios a lo que muchas empresas se refieren a la hora de buscar análisis de inmuno diagnóstico o microbiológicos.

Según indagaciones hechas en motores de búsqueda por internet, tales como Google, se pueden encontrar referencias del LAMIR en diversos

directorios de laboratorios de empresas, así como en trabajos de investigación y tesis realizadas en dicha entidad.

Por lo anterior se infiere que el LAMIR opera sin necesitar un sistema publicitario de ninguna índole; no obstante, el LAMIR ha considerado en un futuro en realizar acciones destinadas a publicitar y dar a conocer el laboratorio para ofrecer su servicio a más empresas, personas particulares u otras entidades.

1.3. El proceso de inmuno-diagnóstico

Es una metodología que se utiliza para obtener un informe en el cual se basaran los programas sabiendo que no habrá ningún colapso en el sistema, ya que previo a utilizarlo se han hecho las muestras necesarias para que sea sólido y eficaz.

El LAMIR divide sus actividades de en: ingreso y recepción de muestras. En el LAMIR se reciben las muestras de distinto tipo, intensidad en un formulario donde se pide a la empresa o persona quien lo posee. El tipo de análisis que se solicita, la persona quien las ingresa, entre otros datos.

1.4. Análisis de la muestra

Con los instrumentos del que dispone el laboratorio, se realiza el análisis solicitado.

Realización del informe: se tabulan los resultados y se realiza el informe de dicho análisis.

En el LAMIR se reciben las muestras de distintos tipos, entre están superficies, agua potable o purificada.

Los tipos de exámenes que maneja el LAMIR para sus operaciones son:

- Análisis microbiológico de agua (potable)
- Análisis microbiológico de agua (purificada)
- Informe de aislamiento e identificación de shigella en alimentos
- Informe de aislamiento e identificación de coliformes y *Escherichia coli* en alimentos
- Informe de aislamiento e identificación de salmonella en alimentos
- Informe de aislamiento e identificación de *Staphylococcus aureus* en alimentos y productos farmacéuticos
- Informe de coliformes totales o fecales (agua)
- Informe de higiene de manos
- Informe de higiene de Superficies
- Informe de pruebas de esterilidad
- Informe de recuento aeróbico de Hongos
- Informe de recuento de coliformes e identificación de *Echerichia coli* (agua purificada)
- Recuento aeróbico de bacterias (agua potable)
- Recuento de coliformes e identificación de *Escherichia coli* (agua potable)
- Recuento aeróbico de bacterias (agua purificada)
- Límite microbiano en ambientes
- Recuento heterotrófico en placa de bacterias, coliformes y E. Coli
- Informe de límite microbiano en ambientes

1.5. Elaboración de diagnósticos

Según observaciones y conversaciones entabladas con personal del LAMIR, se determinó que para elaborar diagnósticos se necesitarían elaborar los siguientes módulos:

- Elaboración de informes de resultados
- Registro y consulta de informes
- Impresión de informes
- Registro y control de equipo del laboratorio

2. MARCO TEÓRICO

La primera teoría sobre el software fue propuesta por Alan Turing en su ensayo de 1935 sobre números computables, con una aplicación destinada a la toma de decisiones. El término software fue utilizado por primera vez de forma escrita por John W. Tukey en 1958. El estudio de los campos académicos sobre el software se divide en informática y la ingeniería de software.

Como los programas cada vez entraban más en el reino de firmware y el hardware por si sólo se hacía más pequeño, más barato y más rápido debido a la Ley de Moore, los elementos de la computación que primero se consideraban software, pasan a ser hardware.

La mayoría de las compañías de hardware hoy en día tienen más programadores de software en nómina que diseñadores de hardware, ya que las herramientas de software han automatizado muchas de las tareas de los ingenieros en circuitos. Al igual que la industria automotriz, la industria del software ha crecido de unos pocos visionarios que operaban en su garaje con sus prototipos.

Steve Jobs y Bill Gates fueron los Henry Ford y Chevrolet Luis de sus tiempos. En el caso del desarrollo de software, el despegue final es generalmente aceptado que se produce con la publicación en la década de 1980 de las especificaciones para el IBM Personal Computer. Hoy su movimiento sería visto como un tipo de *público-sourcing*.

Hasta ese momento, el software se incluye con el hardware de los fabricantes de equipos originales (OEM), tales como Data General, Digital *Equipment* y de IBM. Cuando un cliente compra una minicomputadora, estas empresas de informática de hardware, no sólo incluyen sus paquetes de software, sino que también asesoren sobre la ubicación de los equipos normalmente en un espacio refrigerado llamo sala de ordenadores.

La mayoría de las empresas tenían su software en su contabilidad valorados 0 ya que no podían venderlo. Cuando Data General ingresó su software Data General Nova, una compañía llamada *Digidyne* intentó instalar este software que ya había adquirido en un equipo distinto. Data General se lleo a dar licencia para poder hacerlo y fueron a los tribunales.

La Corte Suprema dijo que si *Digidyne* había pagado era propietaria de ese software debía poder instalarlo en el equipo que quisiese, lo que se llamó *Digidyne v.* Poco después IBM publicó los registros de DOS y nació Microsoft. La decisión de la Corte Suprema permitió valorar el software, patentarlo y comerciar con él. Es difícil imaginar hoy que una vez la gente sentía que el software no valía nada sin una máquina.

Hay muchas empresas de éxito hoy en día que venden solo productos de software, aunque todavía hay muchos problemas comunes de concesión de licencias de software debido a la complejidad de los diseños y documentación, lo que lleva a los controles de patentes.

Con las especificaciones de software de código abierto y la posibilidad de concesión de licencias de software, nuevas oportunidades se levantaron de herramientas de software que luego se convirtieron en el estándar de facto, como DOS para los sistemas operativos, sino también diversos programas de

procesamiento de texto y hojas de cálculo. En un patrón de crecimiento similar, los métodos de desarrollo de propiedad que se convirtió en la metodología estándar de desarrollo de software.

A continuación se describe técnicamente lo que se tomó en cuenta a la hora de realizar el proyecto.

2.1. Consideraciones técnicas

Entre las consideraciones técnicas que se hicieron para la elaboración de dicho proyecto, se destacan las más importantes, las que son necesarias para que todo el proyecto sea eficiente.

2.1.1. Nivel de visibilidad

Se refiere al perfil de visibilidad que se desea del sistema. En el caso del LAMIR, se pudo pidió que solamente pudiera verse en el laboratorio y por personal del laboratorio, por tanto, se determinó que el nivel idóneo de visibilidad era un nivel privado.

2.1.2. Software

El software considerado para esta aplicación es el software para desarrollo, herramientas para navegación y el sistema operativo a utilizar, y las dependencias.

- Mozilla Firefox: es un navegador web libre y de código abierto descendiente de Mozilla Application Suite y desarrollado por la Fundación Mozilla. Es multiplataforma, estando disponible para varios

sistemas operativos como Microsoft Windows, GNU/Linux, Mac OS X y no oficialmente, FreeBSD18 así como otras plataformas.

- ASP.NET 2008: es un framework para aplicaciones web desarrollado y creado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el *Common Language Runtime*, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.
- AJAX: acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

- Visual Studio 2008: Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET.
- Javascript: es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, con un leve tipado dinámico.

- **SQLServer 2005:** es una plataforma global de base de datos que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial (BI). El motor de la base de datos SQL Server 2005 ofrece almacenamiento más seguro y confiable tanto para datos relacionales como estructurados, lo que le permite crear y administrar aplicaciones de datos altamente disponibles y con mayor rendimiento para utilizar en su negocio.

2.1.3. Sistema operativo anfitrión

Como sistema operativo anfitrión se determinó que la aplicación sería utilizada en sistemas Windows XP, Service Pack 2 y Windows 7 Ultimate Edition, siendo que no hubo una compilación que funcionara para ambos sistemas, por lo que se determinó que se hicieran las respectivas compilaciones para cada una de los sistemas operativos.

2.1.4. Dependencias

Las dependencias son librerías o programas necesarios para el correcto funcionamiento de una aplicación. Así, una aplicación para visualizar vídeos de Youtube, tiene como dependencia el plugin de flash player para visualizar los vídeos.

- **IIS 4.0:** es un servidor web con servicios para el sistema operativo Microsoft Windows integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional para mostrar los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

- About Suite: es una serie de librerías que contienen controles para aplicaciones de ASP.NET. Las mismas están basadas en su mayoría en AJAX y en definiciones ASPX, las cuales permiten definir una serie de controles totalmente compatibles con los navegadores más comunes, y a la vez ofrecen una fácil interfaz para adaptar dichos controles a cualquier página ASP.

2.1.5. Hardware

En español se le conoce como: toda la parte tangible de un equipo de cómputo, teclado, monitor y otros. A continuación se describen los requerimientos mínimos que debe tener el equipo para que el sistema tenga una función eficiente.

- Procesador: para que funcione correctamente la aplicación sus requerimiento mínimo es un, Pentium IV, de 3 GHZ, con 1 GB de Memoria RAM como mínimo. Recomendable sería un Procesador Dual Core de 2.5 GHz con 2 GB de memoria RAM e ideal sería un procesador Core 2 Duo, con 3 GB de RAM.
- Disco Duro: como mínimo, se necesitan 500 MB de almacenamiento para el correcto funcionamiento de la aplicación. Recomendado serían 2 GB e ideal serían 4GB o más.

2.1.6. Factores de calidad

Estos son necesarios para que los usuarios que utilizaran dicho programa estén seguros que no habrá ningún fallo al momento de operarlo, entre estos se destacan los siguientes:

- Usabilidad: en cuanto al tipo de aplicaciones y sistemas operativos a los que estaba acostumbrado el personal del LAMIR, se determinó que la mayoría estaba solamente familiarizada con sistemas Windows y aplicaciones de ofimática tales como Microsoft Word y páginas web; además que el equipo disponible para alojar la instalación tenía un sistema operativo Windows; por tal motivo se determinó que la aplicación debía estar diseñada para un sistema Windows y su forma de uso debía ser muy similar a la de una aplicación de Microsoft Office y con un protocolo de ingreso de información similar a los sitios web.
- Seguridad: los aspectos que tuvieron que tomarse en cuenta para el aspecto de la seguridad fueron: la cantidad de público o nivel de visibilidad que se requería para la aplicación, el tipo de aplicaciones y sistemas operativos con los cuales el personal del LAMIR estaba familiarizado así como la usabilidad del mismo.

El inventario de equipo con el que se contaba, las especificaciones técnicas y características del equipo donde se instalaría la aplicación y la compatibilidad de este con las aplicaciones y el rendimiento de las mismas.

Según lo indicado por las autoridades del LAMIR, necesitaban una aplicación que fuera única y exclusivamente para uso interno y que solamente personal de la entidad tuviera acceso a ella. Por lo tanto el nivel de visibilidad requerido era nulo y dicho sistema debía cumplir con restricción de seguridad.

Otros factores de calidad: en cuanto a los factores de calidad, se determinó que el aspecto de la compatibilidad y el rendimiento era uno de

los más importantes para lograr un correcto desempeño de la aplicación, por lo cual se hicieron diversas evaluaciones de rendimiento de aplicaciones de prueba diseñadas para diversos tipos de sistemas operativos.

2.2. Consideraciones técnicas del desarrollo del software

Para la elaboración de la aplicación se eligió como aplicación de desarrollo a VisualBasic.Net y VisualBasic para desarrollar la aplicación. Posteriormente se determinó que serían necesarias una serie de pruebas y evaluaciones para determinar el correcto funcionamiento de dicha aplicación.

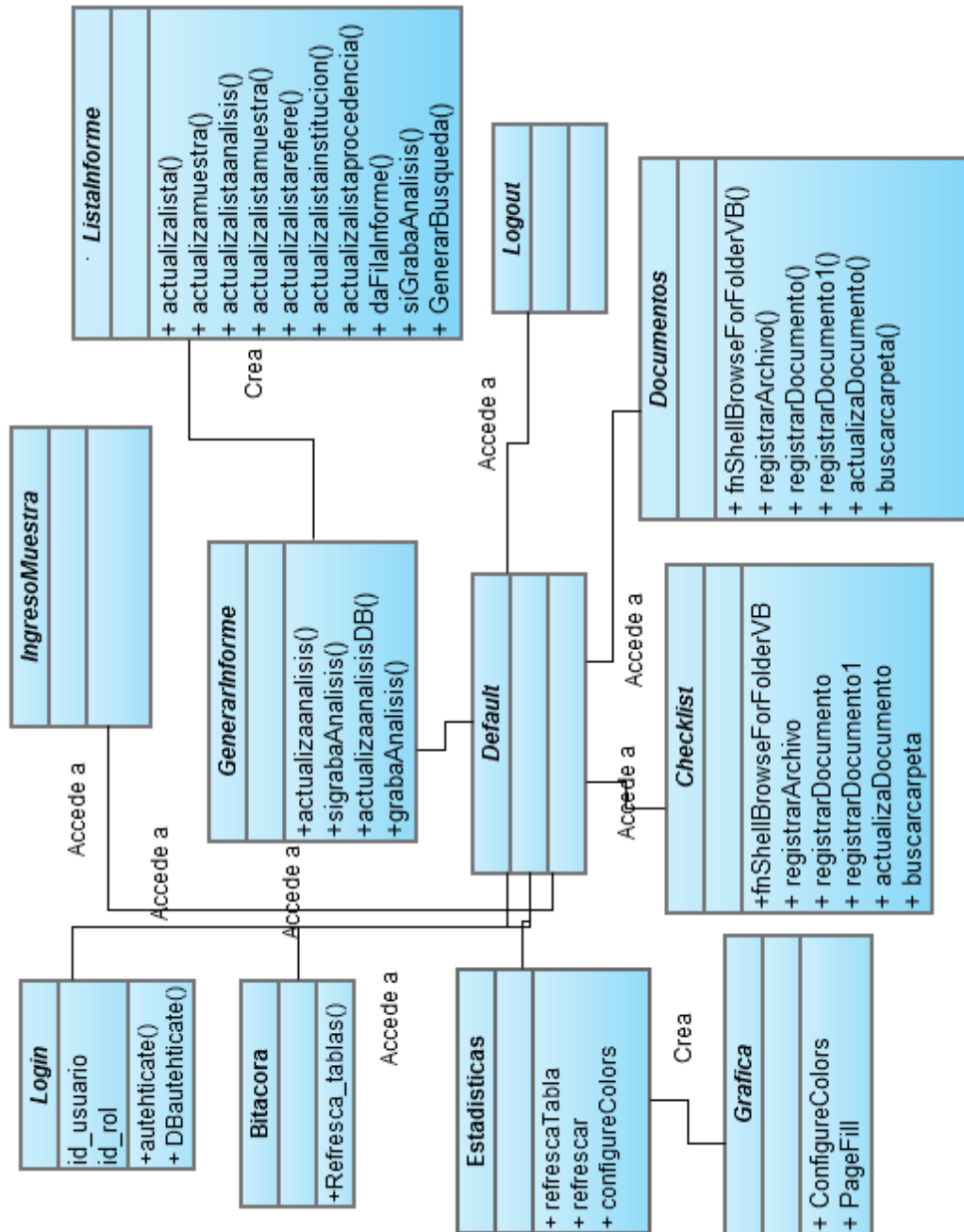
Debido a que el equipo de cómputo para el desarrollo de la aplicación tenía un sistema operativo Windows 7 y aunque el sistema operativo huésped era Windows XP y estaba basado en una arquitectura de 32 bits, se elaboraron pruebas de aplicaciones compiladas en sistemas Win 7, siendo funcional en otros equipos con el mismo sistema operativo; sin embargo en el sistema anfitrión la aplicación no pudo ejecutarse.

Por tal motivo, se determinó instalar Virtualbox, que es una herramienta gratuita de virtualización la conocida empresa Oracle, para crear una máquina virtual con el sistema operativo Windows XP, para compilar los archivos ejecutables de la aplicación sin que existieran problemas de compatibilidad.

2.2.1. Diseño de módulos (diagrama Entidad-Relación)

Para dar una mejor visión del *software* se procede a demostrar un diagrama de Entidad-Relación, en donde se podrá observar claramente la lógica del negocio y la utilidad del mismo.

Figura 1. Diagrama Entidad-Relación de los módulos



Fuente: elaboración propia.

3. PRELIMINARES

Entre los preliminares que se realizaron están los siguientes:

3.1. Entrevistas

Las charlas empezaron en febrero del 2013, para tomar idea de qué estaban necesitando utilizar, qué querían hacer y elaborar, qué tipo de aplicación iban a necesitar, qué tamaño y capacidad requeriría, tanto de funcionalidades como de carga de trabajo.

El primer punto a tomar en cuenta fue negociar la cantidad de entregables y el tiempo estipulado. Se requería de parte del LAMIR un proyecto con diversas funcionalidades en un tiempo de tres meses, siendo necesarios para un proyecto de ese tamaño al menos 6 meses o incluso más.

Se pudo explicar eso con éxito y se acordó una serie de requerimientos que eran fundamentales para ese tiempo y dejando los otros para un tiempo posterior, situación con la que se tuvo problemas dado los diversos inconvenientes de usabilidad que al final terminaban afectando la robustez del proyecto, en cuanto a coherencia de datos, ya que la forma de ingresar, que al inicio tuvo una elevada complejidad, hacía que el ingreso de los datos afectaba directamente con la visualización de los mismos, y algunos otros concernientes concerniente a esos temas.

3.2. Toma de requerimientos

Para la toma de requerimientos se realizaron una serie de entrevistas en el mes de marzo del 2013, con el objetivo de determinar los entregables más importantes y crucial es para dicho proyecto. Asimismo se realizaron diversas versiones de prueba con la finalidad de entregar una versión para que en el LAMIR observaran e interactuaran con los posibles entornos e interfaces.

Un punto difícil en dichas entrevistas fue el entendimiento en la comunicación y la forma de la misma. En dicho intercambio de información no existió una aclaración puntual acerca de que las primeras entregas eran prototipos para mostrar funcionalidad, no que los prototipos en sí serían los entregables, algo que provocó no pocos inconvenientes y molestias por parte de los clientes ya que, en muchas ocasiones la cantidad de cambios era sustancial y profunda situación que requería tiempo y, al final, atrasó los resultados en base a las expectativas.

Otro punto muy difícil de sobrellevar fue el error a la hora de efectuar el cronograma. Muchos atrasos se suscitaron por deficiencias en la forma de explicarles el proceso del proyecto, lo que generaba inconformidad y problemas respecto a la usabilidad del proyecto y los tiempos de entrega, así como la facilidad, algo que ha terminado por ser algo prioritario para los usuarios del LAMIR, aparte de la robustez y la funcionalidad, que es un requisito indispensable *per se*.

Se realizó una toma de requerimientos pensando en utilizar un sistema de trabajo basado en metodologías ágiles, tomando elementos de las mismas que más se adecuaban a lo que se pretendía trabajar.

Por tanto la toma de requerimientos fue puntual, sin ahondar demasiado en los detalles, tomando objetivos y productos, entregables, que se requerían, para luego trabajar en un prototipo, mostrar los resultados y funcionalidad y, con base en lo que opinaran en el LAMIR, redefinir el requerimiento, dividirlo en varios o modificarlo.

3.3. Negociaciones de la aplicación

Después de las entrevistas y toma de requerimientos se terminaron de definirlos entregables, siendo estos al inicio, 8, que se terminaron extendiendo a 10 por la importancia de otras funcionalidades, como la bitácora del sistema o el módulo de seguridad para modificar información, que eran cruciales para el proyecto.

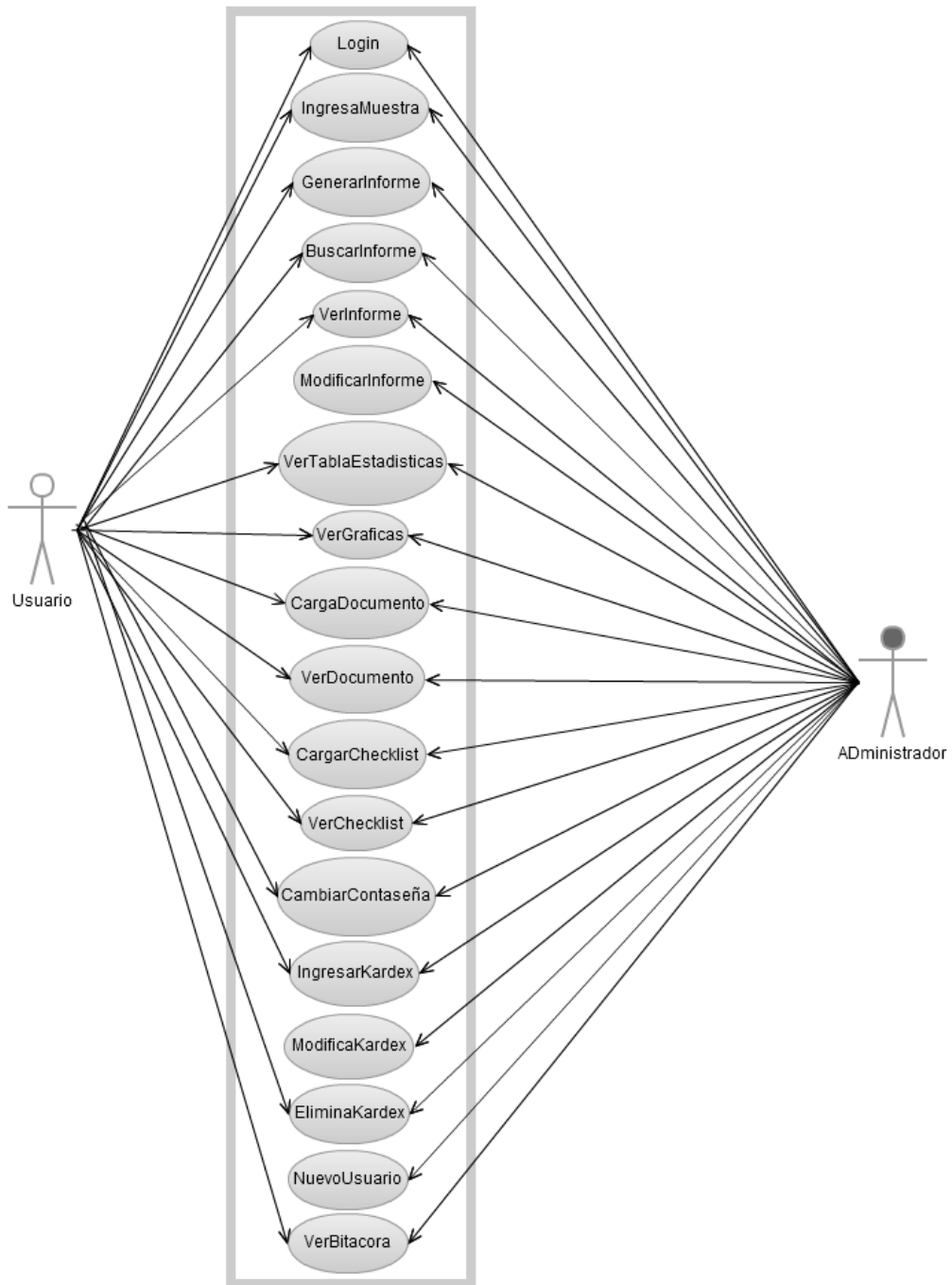
Para robustecer el hardware con que contaba el LAMIR y para el alojamiento del sistema, se invertirían un total de Q2 000.00 en instalación de equipo más Q10 000,00 en la instalación en una nueva computadora para instalación del equipo.

Se realizaron gestiones de diversa índole para obtener el mejor equipo a disposición, pero dado que la obtención de equipo nuevo tenía un costo mayor del que disponía el LAMIR para esa operación, se optó por mejorar el equipo con el que ya contaba el LAMIR para hacer la instalación del software.

3.4. Casos de uso

Se desea demostrar en la figura 2, como será la interacción del usuario, el *software* y el administrador del mismo. Se demuestran las escalas en orden descendente de los pasos que deberán seguir para poder ingresar.

Figura 2. **Casos de Uso de cada uno de los módulos**



Fuente: elaboración propia.

4. ANÁLISIS DE LA SOLUCIÓN

Se realizó el análisis de la solución y se tomarón en cuenta las siguientes partes:

4.1. Consideraciones técnicas

A continuación se expondrá una breve explicación del por qué se usan las herramientas para el desarrollo del software LAMIR, luego se explicará el porqué del uso de las metodologías de desarrollo utilizadas.

4.1.1. .NET

La herramienta que se eligió para desarrollar la aplicación fue .NET, la cual es una herramienta de desarrollo de software diseñada por la empresa Microsoft. Es un *framework* de Microsoft que funciona con independencia de plataforma de hardware y facilita el desarrollo de aplicaciones.

Su propósito es ofrecer una manera económica, segura, veloz y robusta, de desarrollar aplicaciones permitiendo integrar en poco tiempo soluciones entre empresas con acceso simple y sin problemas de compatibilidad a todo tipo de información desde cualquier tipo de dispositivo.

La herramienta presenta las siguientes utilidades:

- Permite desarrollar sitios web basados en ASP o ASPX

- Comptabilidad con herramientas para elaborar páginas dinámicas tipo AJAX y Javascript, entre otros
- Permite conexión a base de datos Mysql, Access, SQLServer, Oracle, entre otras
- Provee controles para crear páginas con múltiples funcionalidades y e integración con librerías de compatibilidad, y provee flexibilidad para instalar librerías extras

La herramienta .NET es utilizada para crear el programa de visualización de las pantallas del proyecto, aparte que fue la plataforma desde la que se creó las bases de datos y el programa ASP.NET, fue utilizada para crear las páginas y clases para el funcionamiento de la herramienta.

4.1.2. SQL Server

Es un sistema de Base de Datos creado por Microsoft. Sus lenguajes para creado como una respuesta a otros sistemas de la competencia como Oracle, MySQL, entre otros.

El usar SQL Server presenta los siguientes beneficios:

- Funciones en el desarrollo de la aplicación y soporte de transacciones
- Soporta procedimientos almacenados
- Incluye también un entorno gráfico para administración, que permite el uso de comandos DDL y DML de forma gráfica
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y los terminales o clientes acceden a la información
- Además permite administrar información de otros servidores de datos.

4.2. Explicación de la arquitectura de software

Al hacer la arquitectura de software se determinó que la mejor opción consistía en elaborar una aplicación de escritorio, considerando solo una computadora y como metodología de desarrollo a Extreme Programming (XP).

4.2.1. Aplicación de escritorio

En una aplicación de escritorio normalmente no se inicia sesión por cada aplicación que se usa, sólo se inicia sesión una vez cuando se pone a funcionar el sistema operativo, asumiendo que se va a abrir una aplicación para ver nuestra lista de tareas:

- El usuario carga la aplicación
- La aplicación (el código), se conecta a la base de datos y recupera la información del usuario
- La aplicación muestra al usuario la información solicitada

4.3. ¿Por qué usar Extreme Programming (XP)?

Es una metodología de desarrollo ágil. Es el más destacado de los de desarrollo de software. Al igual que estos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida de proyecto es una aproximación mejor y más realista

que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

4.3.1. Pros y contras de la metodología

Como en todo *software*, máquina, diseño, existen ventajas y desventajas, las cuales deben ser tomadas en cuenta. En esta para se especifican los pros y los contras de utilizar dicho *software*.

- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas: es recomendado que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- Integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los fallos, antes de añadir nueva funcionalidad. Hacer entregas frecuentes.

- Refactorización del código, es decir escribir ciertas partes del código para aumentar su legibilidad y capacidad de mantenimiento pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad del código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.
- Otra ventaja es que esta metodología es la que mejor se adapta a grupos de trabajo de pocas personas, siendo ideal y fácil su utilización para que una sola persona desarrolle una aplicación.

4.3.2. Forma de implementación

Para la implementación de la misma, se usaron los 5 valores de la metodología *Xtreme Programming*, con las limitaciones de ser solo una persona quien programó la aplicación y no dos, que es lo comúnmente usado en este tipo de aplicación.

- Simplicidad: es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento.
- Comunicación: el código comunica mejor cuanto más simple sea. Los programadores se comunican constantemente gracias a la programación por parejas.

La comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo. El cliente decide qué características tienen prioridad y siempre debe estar disponible para solucionar dudas.

- Retroalimentación (*feedback*): Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante.
- Coraje y valentía: muchas de las prácticas implican valentía. Una de ellas es siempre diseñar y programar para hoy y no para mañana. Esto es un esfuerzo para evitar empantanarse en el diseño y requerir demasiado tiempo y trabajo para implementar todo lo demás del proyecto.
- Respeto: los miembros del equipo se respetan los unos a otros, porque los programadores no pueden realizar cambios que hacen que las pruebas existentes fallen o que demore el trabajo de sus compañeros. Los miembros respetan su trabajo porque siempre están luchando por la alta calidad en el producto y buscando el diseño óptimo o más eficiente para la solución a través de la refactorización del código.

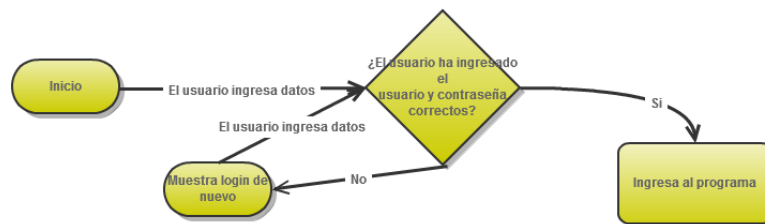
Los miembros del equipo respetan el trabajo del resto no haciendo menos a otros, una mejor autoestima en el equipo y elevando el ritmo de producción en el equipo.

5. DESARROLLO DE LOS MÓDULOS

5.1. Login

El usuario visualiza una pantalla donde se le solicita ingresar usuario y contraseña para ingresar a la plataforma. La aplicación deberá informar sino tuvo éxito el acceso a la plataforma. En ese caso, proporcionar el ingreso de nuevo.

Figura 3. Diagrama de flujo

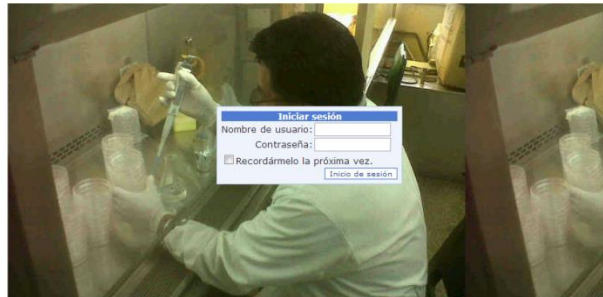


Fuente: elaboración propia.

Se establecieron conexiones a la base de datos y los objetos de manipulación de datos denominados *DataSet* para conectar con la base de datos y así realizar las inserciones de registros por cada muestra ingresada al sistema.

También se hicieron pruebas con los campos de la base de datos y que los mismos coincidieran con los datos ingresados inicialmente, evitando así inserción de caracteres no ingresados en los campos de ingreso de la interfaz de la plataforma.

Figura 4. **Captura en pantalla de la versión definitiva del módulo Login en tiempo de ejecución**



Fuente: elaboración propia.

5.2. Ingreso de muestras

El usuario accede a la pantalla donde se solicitan los datos para ingresar la muestra. La aplicación debe mostrar si el ingreso de la muestra tuvo resultado. En caso contrario, deberá mostrar la misma página para que intente de nuevo.

Figura 5. **Diagrama de flujo para el módulo de ingreso de muestras**



Fuente: elaboración propia.

Se establecieron conexiones a la base de datos y los objetos de manipulación de datos denominados *DataSet*, para conectar con la base de datos y así realizar las inserciones de registros por cada muestra ingresada al sistema.

También se hicieron pruebas con los campos de la base de datos y que los mismos coincidieran con los datos ingresados inicialmente, evitando así inserción de caracteres no ingresados en los campos de ingreso de la interfaz de la plataforma.

Figura 6. **Versión final del módulo de ingreso de muestras**

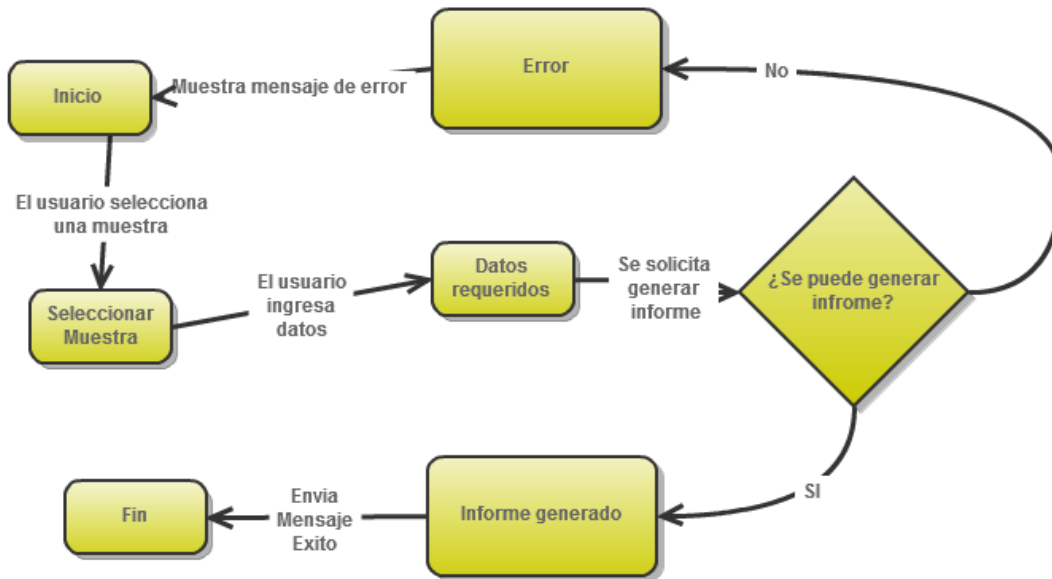
<input type="text"/>	<p>Fecha</p> <p>Ingreso <input type="text"/></p> <p>Muestreo <input type="text"/></p> <p>Hora <input type="text"/></p>
<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	<input type="checkbox"/> Análisis microbiológico de agua (potable) <input type="checkbox"/> Análisis microbiológico de agua (purificada) <input type="checkbox"/> Aislamiento e identificación de Shigella en alimentos <input type="checkbox"/> Aislamiento e identificación de Coliformes y Escherichia coli en alimentos <input type="checkbox"/> Aislamiento e identificación de Salmonella en alimentos <input type="checkbox"/> Aislamiento e identificación de Staphylococcus aureus en alimentos y productos farmacéuticos <input type="checkbox"/> Coliformes Totales o Fecales (agua) <input type="checkbox"/> Higiene de manos <input type="checkbox"/> Higiene de superficies <input type="checkbox"/> Pruebas de esterilidad <input type="checkbox"/> Recuento aeróbico de hongos <input type="checkbox"/> Recuento de coliformes e identificación de Escherichia coli (agua purificada) <input type="checkbox"/> Recuento Aeróbico de Bacterias (Agua Potable) <input type="checkbox"/> coliformes e identificación de Escherichia coli (agua potable) <input type="checkbox"/> Aeróbico de bacterias (agua purificada) <input type="checkbox"/> límite microbiano en ambientes <input type="checkbox"/> Recuento heterotrófico en placa de bacterias, coliformes y E.Coli <input type="checkbox"/> Recuento de Coliformes, Escherichia coli y Salmonella

Fuente: elaboración propia.

5.3. Generar informe

La aplicación muestra una interfaz para seleccionar muestras y análisis ya ingresados para generar un informe del mismo. A su vez, el programa permite ingresar ciertos datos necesarios para generar dicho informe.

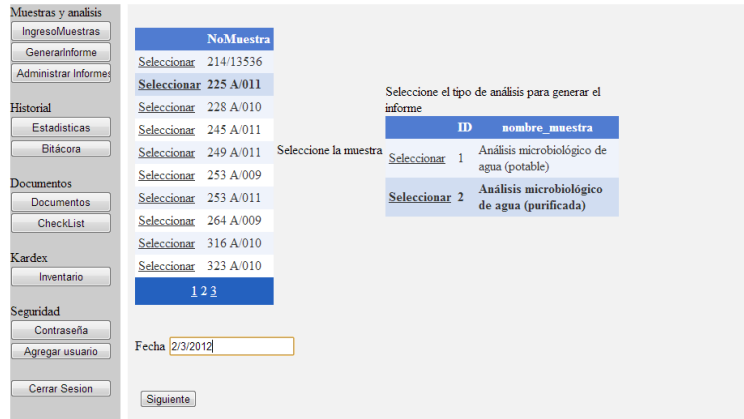
Figura 7. Diagrama de flujo del módulo de generación de informes



Fuente: elaboración propia.

Se hicieron pruebas de varias librerías, desde la utilización de la librería TinyMCE, hasta de *plugins* con controles especializados para utilizar, tal es el caso de los objetos del *Obout Suite* para ASP.NET.

Figura 8. **Versión final del módulo para generar informe**

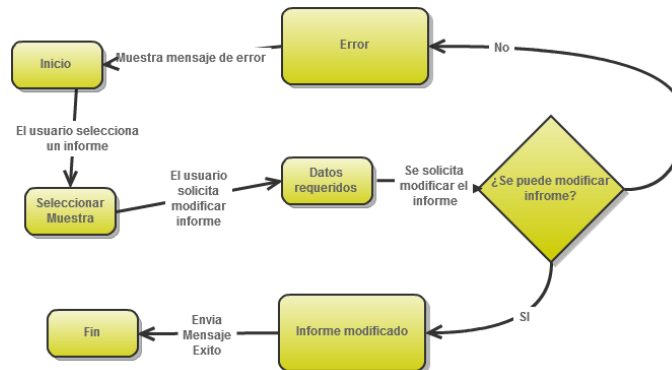


Fuente: elaboración propia.

5.4. Administrar informes

Como en todo *software online* siempre habrá un administrador. La aplicación permitirá al usuario seleccionar un informe y modificar cualquier parte del mismo sin que tenga alguna restricción

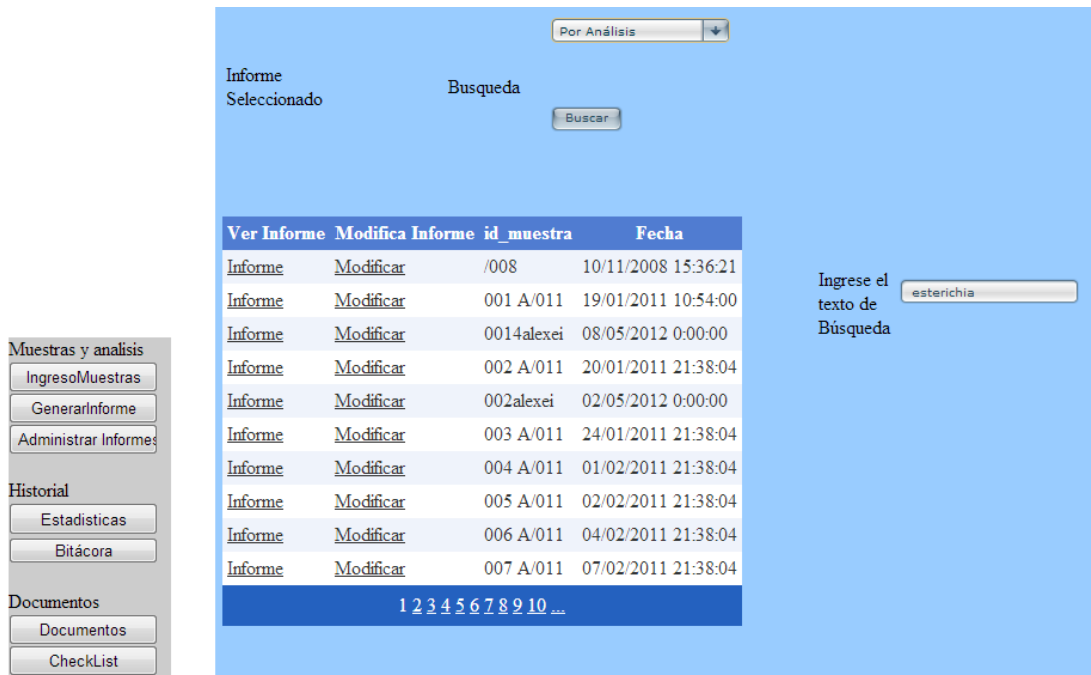
Figura 9. **Diagrama de flujo del módulo de modificación de informe**



Fuente: elaboración propia.

Se requirió de múltiples pruebas con librerías para lograr un correcto control para mostrar y capturar texto ingresado por el usuario y librerías para controles Grid, necesarios para seleccionar el registro a ver o modificar.

Figura 10. **Versión final del módulo para administrar informes**

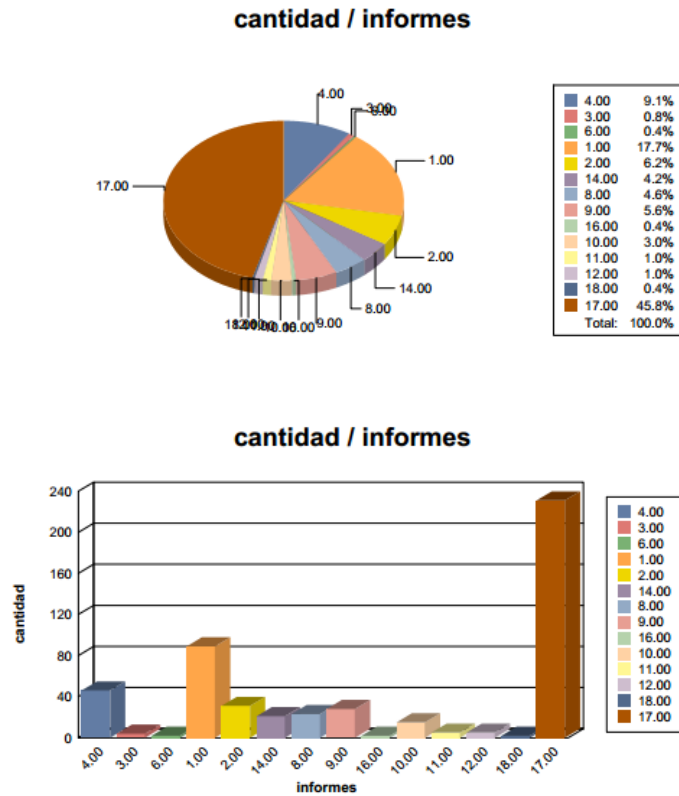


Fuente: elaboración propia.

5.5. Estadísticas

Este módulo permite revisar las estadísticas de las muestras ingresadas, divididas por tipos de muestras y mostradas en tabla y por gráfica de histograma de frecuencias.

Figura 11. Versión final del módulo de estadísticas



Fuente: elaboración propia.

Se tuvo que subsanar ciertas discrepancias en cuanto al código necesario para insertar la librería *webchart*, pero a base de pruebas y ensayos del aspecto del *webchart*, se logró insertar de manera aceptable y funcional.

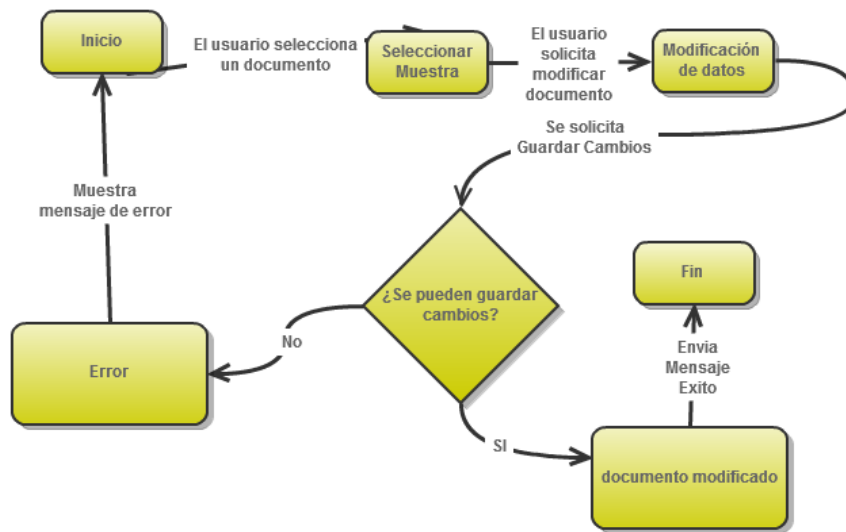
5.6. Seguridad

El módulo permite cambiar de contraseña, solicitando primero la contraseña actual para propiciar que solamente la persona que ingresa con su cuenta de usuario pueda cambiar su contraseña.

5.7. Documentos

Después de una investigación profunda y analizada detenidamente, se llego a la conclusión que la aplicación debe permitir editar documentos guardados y almacenados previamente.

Figura 12. Diagrama de flujo del módulo de edición de documentos



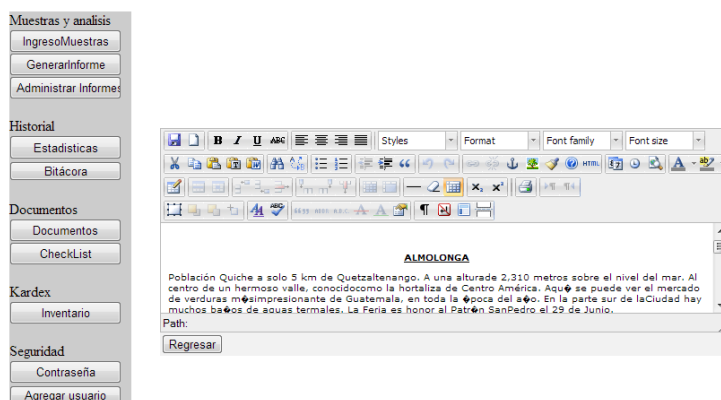
Fuente: elaboración propia.

Se utilizaron objetos y controles para búsqueda y desarrollo de dicho módulo; sin embargo, los controles utilizados no satisfacían los requerimientos solicitados para la búsqueda de documentos.

Se utilizaron controles para búsqueda de la librería *Obut Suite* y con ella se lograron cargar los archivos.

Al momento de registrarse un documento este genera un nuevo registro en la tabla de documentos de la base de datos y debería ser visible desde el control utilizado para abrir los documentos.

Figura 13. **Versión final del módulo documentos**

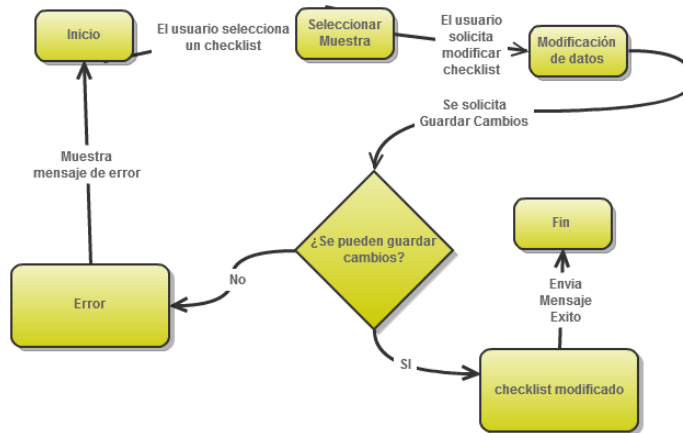


Fuente: elaboración propia.

5.8. Desarrollo del módulo de *checklist*

Es la parte en donde el usuario puede ingresar y modificar campos personales de su cuenta. La aplicación permite editar *checklist*, guardados y almacenados previamente.

Figura 14. Diagrama de flujo del módulo *checklist*



Fuente: elaboración propia.

Se realizaron pruebas de grabación de datos y apertura de *checklist* usando la librería pdfcreator y el ejecutor de comandos por llamada a consola que trae Visual Studio respectivamente.

Figura 15. Versión final del módulo *checklist*



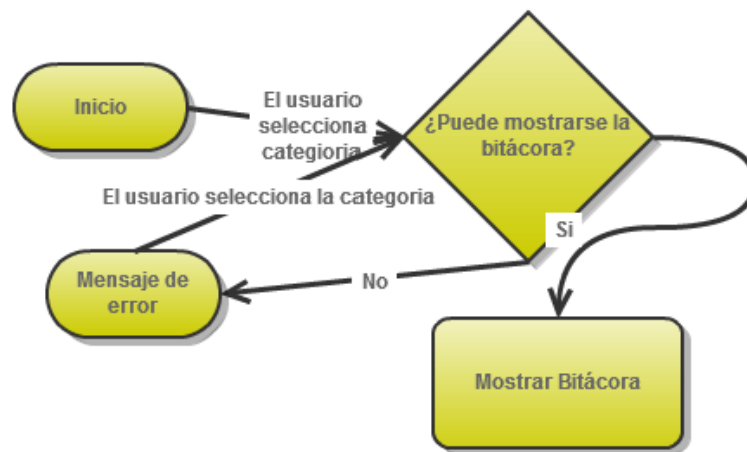
Fuente: elaboración propia.

5.9. Desarrollo del módulo de bitácora

Este módulo fue creado para mostrar un registro de todas las actividades realizadas en la aplicación, describiendo la acción realizada y el usuario que la realizó.

La aplicación permite ver una bitácora de cada una de las acciones realizadas en la plataforma, divididas por tipo de acciones e identificando correctamente el usuario que las realizó.

Figura 16. Diagrama de flujo del módulo bitácora



Fuente: elaboración propia.

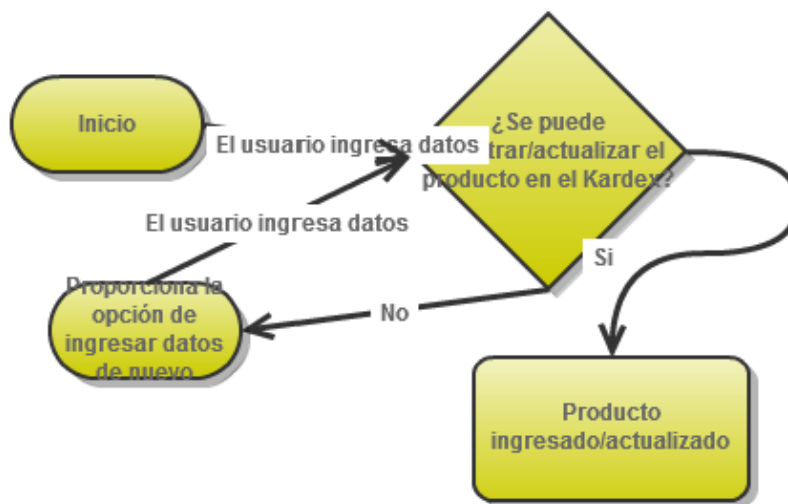
Se consideró la interacción de los usuarios con los controles seleccionados, la carga de datos de los grs con los *dataset* se hizo carga tanto en tiempo real como en tiempo de carga del módulo y también se consideró el refrescado del *grid* con función *databind*.

5.10. Desarrollo del módulo de inventario

Este módulo se creó con el objetivo de permitir que el usuario tipo administrador pueda registrar a más usuarios en el sistema.

La aplicación permite ingresar y registrar equipo e insumos a la base de datos de la misma, la cual llevará control del *kardex*.

Figura 17. Diagrama de flujo del módulo inventario



Fuente: elaboración propia.

Se tuvo que considerar los campos de *password* y la verificación de la existencia de usuarios ya registrados.

6. CAPACITACIÓN

6.1. Plan de trabajo

Para tener una mejor adaptación y manipulación del programa se debe proceder a capacitar y retroalimentar a los usuarios que se encargaran de ejecutarlo, para ello se debe tener en cuenta lo siguiente:

6.1.1. Planificación

Se elaboró un cronograma donde se realizaron sesiones de capacitación, dos por semana, en las cuales se entregó la versión finalizada de la aplicación, susceptible a cambios por sugerencia y peticiones de las personas que utilizaran la aplicación.

Después de las capacitaciones y durante las mismas, se recibieron retroalimentaciones y por las cuales se realizaron cambios a la aplicación.

6.1.2. Realización de la capacitación

Durante cinco semanas se realizaron capacitaciones y a la vez se entregaron vídeo tutoriales para mostrar el uso de cada uno de los módulos y además se resolvieron dudas acerca de la utilización de los mismos.

Las personas capacitadas fueron el personal del laboratorio que se encarga de la recepción de muestras así como quienes se dedican a realizar los informes de las muestras. Adicional a eso se les proporcionó sus credenciales

de autenticación así como a la persona encargada del laboratorio se le dieron las credenciales de autenticación como administrador.

6.2. Finalización de la capacitación y entrega de los productos finales

Para finalizar, se entregaron un CD con el instalador de la aplicación y de los programas y dependencias necesarios para el mismo y un DVD con los vídeos tutoriales para la utilización del mismo. Además se entregó un manual de usuario con dicha aplicación.

6.3. Beneficios de la aplicación

Para convencer al usuario, así como al consumidor final se presentan los beneficios y ventajas que la aplicación ofrece y con esto se espera que el usuario comprenda la necesidad de utilizarla en su ámbito laboral.

- Centralización de actividades: en vez de utilizar varias aplicaciones y software para realizar las tareas de impresión y registro de los informes, ahora se realiza todo en un solo programa que es la aplicación realizada.
- Agilización de los procesos: el hecho que todo se haga en un solo programa y que el mismo haya sido hecho a la medida permite que las actividades se realicen más velozmente.

Permite cumplir los estándares de calidad porque ahora con la creación de dicho programa se pueden cumplir más fácilmente con los estándares de calidad.

- Seguridad: debido a que el programa solicita credenciales de autenticación y que las operaciones del programa están restringidos por rol y permisos, la aplicación ofrece seguridad de datos y privacidad.

CONCLUSIONES

1. Se desarrolló un software referente de los servicios de microbiología del LAMIR y el mismo responde a los requerimientos del mismo.
2. Se considera que el mismo está alineado a los objetivos de dicha entidad.
3. Se crearon módulos que permiten registrar informes y muestras así como almacenarlas e imprimirlas.
4. Se desarrollaron módulos para edición y muestra de documentos.
5. Se desarrolló un módulo de control para crear *checklist* para control de calidad.
6. Se desarrolló un módulo de bitácora de eventos.
7. Se desarrolló un módulo de generación de estadísticas.
8. Se desarrolló un módulo de registro de equipo.
9. Se desarrolló un módulo de cambio de contraseña.
10. Se desarrolló un módulo de creación de nuevos usuarios.

RECOMENDACIONES

1. Al LAMIR: tener a disponibilidad una persona con experiencia en el campo del uso de software para laboratorio para evitar malos entendidos. También se les recomienda hacer inversiones en software y hardware debido a que el disponible es muy limitado.
2. A los nuevos epesistas: un estudio a profundidad de los recursos con los que cuenta la empresa y de los requerimientos que solicitan, así como el tiempo disponible y las exigencias, elaborando un análisis de factibilidad para determinar si dicho proyecto es viable y es posible desarrollarlo con dichas condiciones. Esto es vital.

BIBLIOGRAFÍA

1. Digidyne Solutions. <http://digidynesolutions.com/>. *http://digidynesolutions.com/*. [en línea]. <http://digidynesolutions.com/>. [Consulta: 4 de mayo de 2013.]
2. día, Salud al. saludalia.com. *saludalia.com*. [en línea] http://www.saludalia.com/Saludalia/servlets/contenido/jsp/parser.jsp?nombre=doc_inmunodiagnostico1. [Consulta: 4 mayo de 2013.]
3. Fundación wikipedia. wikipedia.org. *wikipedia.org*. [en línea]. Fundación wikipedia. http://es.wikipedia.org/wiki/Mozilla_Firefox. [Consulta: 3 abril de 2013.]
4. _____. wikipedia.org. *wikipedia.org*. [en línea]. <http://es.wikipedia.org/wiki/ASP.NET>. [Consulta: 6 agosto de 2013.]
5. _____. wikipedia.org. *wikipedia.org*. [en línea]. <http://es.wikipedia.org/wiki/AJAX>. [Consulta el: 2 de mayo de 2013.]
6. _____. wikipedia.org. *Wikipedia.org*. [en línea]. http://es.wikipedia.org/wiki/Microsoft_Visual_Studio. [Consulta: 23 de enero de 2013.]
7. _____. wikipedia.org. *wikipedia.org*. [en línea]. <http://es.wikipedia.org/wiki/JavaScript>. [Consulta: 3 de febrero de 2013.]

8. _____. wikipedia.org. *wikipedia.org*. [en línea]. http://es.wikipedia.org/wiki/Internet_Information_Services. [Consulta: 5 de mayo de 2013.]

9. Microsoft. microsoft.com. *microsoft.com*. [en línea]. <http://www.microsoft.com/spain/sql/productinfo/overview/what-is-sql-server.aspx>. [Consulta: 12 de mayo de 2013.]

APÉNDICES

PROCEDIMIENTO PARA LA INSTALACIÓN DE IIS EN WINDOWS XP PROFESIONAL

Internet Information Server (IIS) es el servidor de páginas web avanzado de la plataforma Windows. Se distribuye gratuitamente junto con las versiones de Windows basadas en NT, como pueden ser Windows 2000 Profesional o Windows 2000 Server, así como Windows XP, también en sus versiones Profesional y Server.

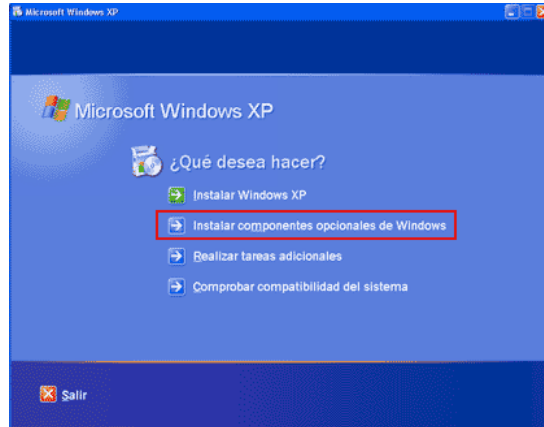
Estas normas de instalación son aplicables, a nivel general, a las que se pueden encontrar en las distintas versiones de los sistemas operativos comentados antes, si bien se ha tomado Windows XP profesional para relatar los pasos y tomar las imágenes de las pantallas.

Agregar componentes adicionales de Windows

IIS se puede encontrar en el propio CD de instalación de Windows XP Profesional. Hay que acceder a la opción de Instalar componentes opcionales de Windows para poder cargarlo en nuestro sistema. Para ello se tienen dos opciones:

- Insertar el CD de instalación de Windows y en la ventana de autoarranque que se muestra, seleccionar la opción que pone Instalar componentes opcionales de Windows.

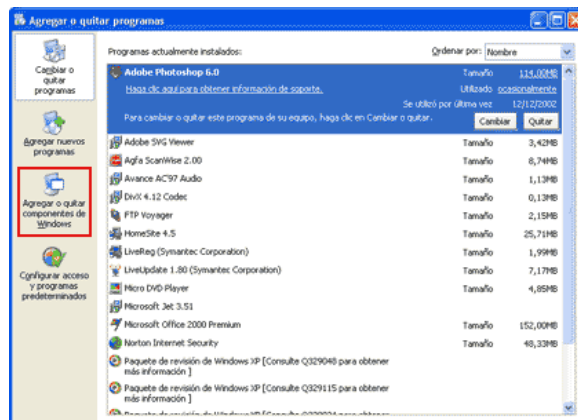
Apéndice 1. Instalación de paquete de Windows XP



Fuente: elaboración propia.

- En el Panel de control, seleccionar la opción de Agregar o quitar programas y en la ventana que sale, pulsar sobre el icono de la izquierda marcado como Seleccionar o quitar componentes de Windows.

Apéndice 2. Ventana de paquetes instalados en Windows

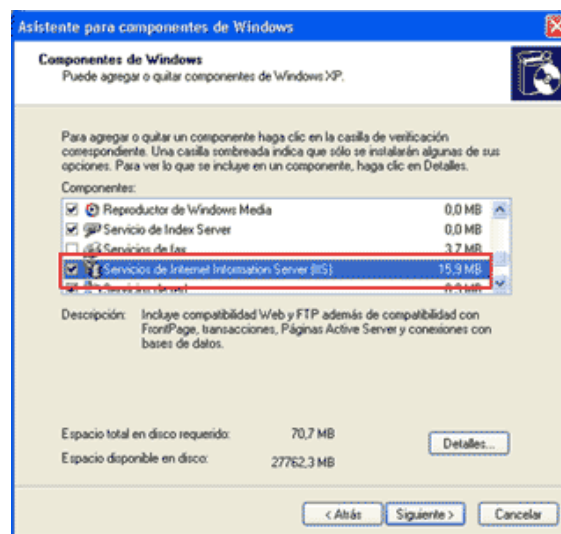


Fuente: elaboración propia.

Ahora muestra la ventana para seleccionar los componentes adicionales de Windows que hay disponibles. En la lista, se marca la opción Servicios de Internet Information Server (IIS). Por defecto se seleccionan unos cuantos componentes, dentro de los que ofrece la instalación de IIS. Se pueden elegir qué componentes se desean instalar presionando el botón marcado como Detalles. Entre los componentes posibles se encuentran las extensiones de Frontpage, documentación, servicios adicionales de IIS, un servidor de FTP (para la transferencia de ficheros con el servidor por FTP), incluso uno de SMTP (para el envío de correos electrónicos).

Si no se sabe qué componentes instalar se pueden dejar las opciones como aparecen en un principio, pues para la mayoría de los casos serán válidas. Sólo un detalle: puede ser adecuado no instalar las extensiones de Frontpage en caso de que no se piense que se vayan a utilizar.

Apéndice 3. Ventana de paquetes instalados en Windows XP



Fuente: elaboración propia.

Una vez instalados los componentes deseados, se puede presionar el botón Siguiente para comenzar la instalación, la cual tardara unos minutos.

Acceder al servidor web

Se puede acceder al servidor web para comprobar si se ha instalado correctamente IIS. Para ello simplemente debemos escribir http://localhost en Internet Explorer y debería aparecer una página web informando que IIS está correctamente instalado. Además, aparecerá la documentación de IIS en una ventana emergente, si es que fue instalada.

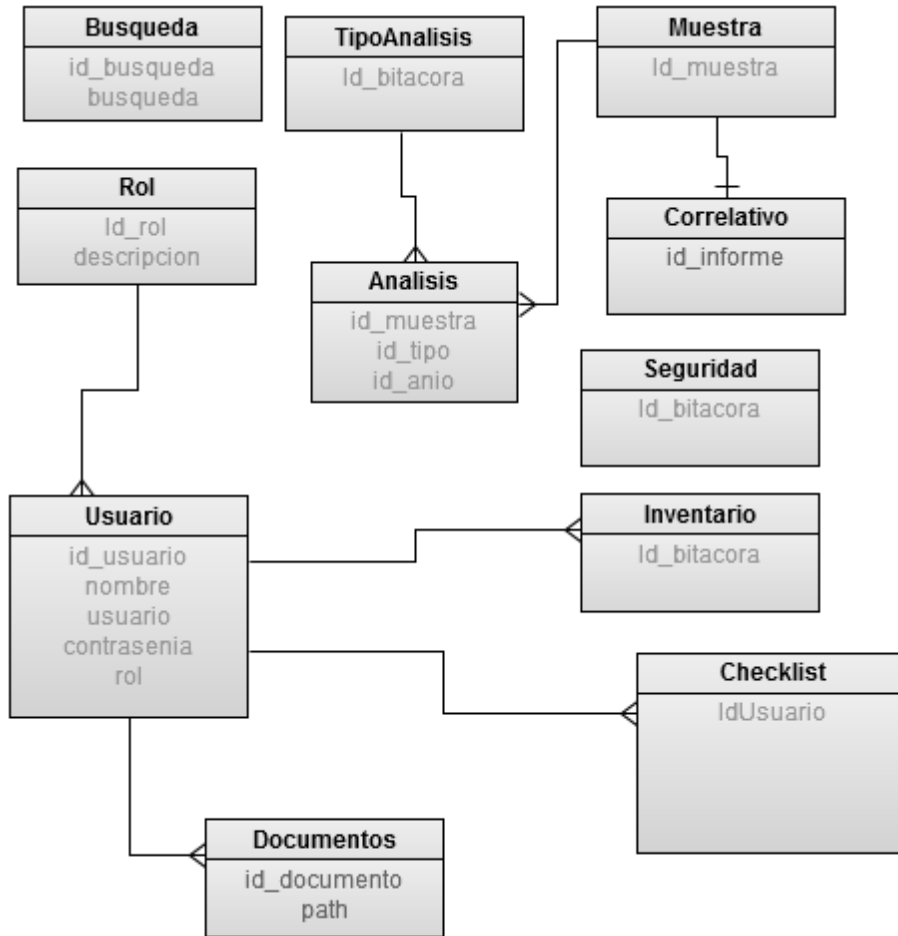
Apéndice 4. Ventana de inicio IIS instalado y funcionando



Fuente: elaboración propia.

DIAGRAMA DE LA BASE DE DATOS

Apéndice 5. Diseño de la base de datos



Fuente: elaboración propia.

DICCIONARIO DE DATOS DE LA BASE DE LA BASE DE DATOS

Usuario: que lleva control del usuario, nombre de cuenta, contraseña del usuario en cuestión.

Apéndice 6. **Tabla de Usuario.**

Usuario	
id_usuario	
nombre	
usuario	
contrasenia	
id_rol	

Fuente: elaboración propia.

Rol: es el rol que desempeña el usuario, que puede ser tipo administrador o usuario ordinario.

Apéndice 7. **Tabla Rol**

Rol	
Id_rol	
descripcion	

Fuente: elaboración propia.

Muestra: en la cual se almacenan todas las muestras ingresadas al sistema.

Apéndice 8. **Tabla muestra**

id_muestra
refiere
institucion
procedencia
fecha_hora_ingreso
fecha_hora_muestreo
hora_ingreso
min_ingreso
hora_muestreo
min_muestreo
id_tipo
ingreso
tipo_muestra
id_usuario

Fuente: elaboración propia.

Análisis: donde se guarda detalle de los tipos de análisis solicitados para la muestra y que posteriormente se les realiza informes.

Apéndice 9. **Detalle de campos de tabla análisis**

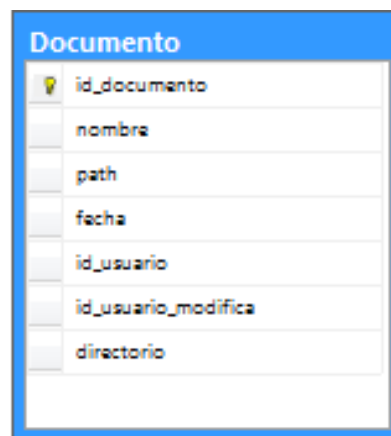
id_muestra
id_tipo
año
informe
resultado
metodologia
conclusion
coninforme
coliformes_totales
coliformes_fecales
presencia_bacterias
presencia_mohos_levaduras
heterotrofico
ambiente
ufm3
fecha_informe
coliformes_totalesT
coliformes_fecalesT
presencia_bacteriasT
presencia_mohos_levadurasT
heterotroficoT
ambienteT
ufm3T
id_usuario
id_usuario_modifica


Fuente: elaboración propia.

Documento

Ésta tabla contiene las direcciones y nombres de los documentos guardados.

Apéndice 10. **Tabla documento**



Documento	
	id_documento
	nombre
	path
	fecha
	id_usuario
	id_usuario_modifica
	directorio

Fuente: elaboración propia.

Inventario

Para el almacenamiento de los componentes y existencia, se creó la tabla inventario.

Apéndice 11. **Tabla inventario**



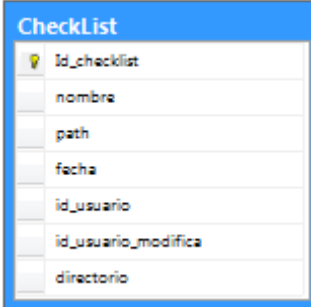
Inventario	
	codigo
	descripcion
	id_usuario

Fuente: elaboración propia.

Checklist

Para el almacenamiento de los documentos se creó la tabla documento, la cual contendrá las direcciones y nombres de los documentos guardados.

Apéndice 12. **Tabla *checklist***



CheckList	
Id_checklist	
nombre	
path	
fecha	
id_usuario	
id_usuario_modifica	
directorio	

Fuente: elaboración propia.

