

**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE QUITO**

**CARRERA:**

**INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:**

**INGENIERO DE SISTEMAS**

**TEMA:**

**ANÁLISIS, DISEÑO Y CODIFICACIÓN DE UN SISTEMA INFORMÁTICO  
PARA GESTIONAR LOS TALLERES DE CAPACITACIÓN OTORGADOS  
POR LA ADMINISTRACIÓN DE LA FEDERACIÓN DE BARRIOS DEL  
NOROCCIDENTE DE QUITO**

**AUTORES:**

**RENATO SANTIAGO DÍAZ SUBIA  
JORGE LUIS CARUA LLUMIGUSIN**

**TUTORA:**

**LINA PATRICIA ZAPATA MOLINA**

**Quito, abril 2016**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros, Renato Santiago Díaz Subia, con documento de identificación N° 1713199550 y Jorge Luis Carua Llumigusin con documento de identificación N° 1716503253, manifiesto nuestra voluntad y cedo a Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado ANÁLISIS, DISEÑO Y CODIFICACIÓN DE UN SISTEMA INFORMÁTICO PARA GESTIONAR LOS TALLERES DE CAPACITACIÓN OTORGADOS POR LA ADMINISTRACIÓN DE LA FEDERACIÓN DE BARRIOS DEL NOROCCIDENTE DE QUITO, mismo que ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en la condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Renato Santiago Díaz Subia  
CC: 1713199550  
Quito, abril de 2016



Jorge Luis Carua Llumigusin  
CC: 1716503253

## DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTORA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Trabajo de titulación, ANÁLISIS, DISEÑO Y CODIFICACIÓN DE UN SISTEMA INFORMÁTICO PARA GESTIONAR LOS TALLERES DE CAPACITACIÓN OTORGADOS POR LA ADMINISTRACIÓN DE LA FEDERACIÓN DE BARRIOS DEL NOROCCIDENTE DE QUITO realizado por Renato Santiago Díaz Subia y Jorge Luis Carua Llumigusin, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, abril 2016.

A handwritten signature in blue ink, appearing to read 'Lina Patricia Zapata Molina', is written over a faint, circular stamp or watermark.

Lina Patricia Zapata Molina  
CI: 0501877278

## **DEDICATORIA**

Dedico este trabajo a Dios y todas las personas que han estado junto a mí de forma incondicional. Agradezco a mis padres, mi esposa e hijos que gracias a su comprensión y apoyo puedo culminar uno más de los objetivos que me trace para la vida.

Renato Santiago Díaz Subia

El presente trabajo está dedicado a Dios, quien me dio fuerza para poder concluir mi carrera, a mis padres por brindarme la confianza, consejos y los recursos para lograrlo, A mi esposa e hijas gracias por estar siempre en esos momentos difíciles brindándome su amor, paciencia y comprensión.

Jorge Luis Carua Llumigusin

## **AGRADECIMIENTO**

A la Universidad Politécnica Salesiana, quién nos ha brindado el conocimiento, la educación y los valores para ser buenos ciudadanos y profesionales con el fin de ayudar con nuestros conocimientos a los demás.

A nuestra tutora de grado Ing. Lina Zapata, por su dedicación, conocimientos y tiempo otorgados para la culminación de nuestra carrera universitaria.

## ÍNDICE

<b>INTRODUCCIÓN</b>	<b>1</b>
<b>CAPÍTULO 1</b>	<b>2</b>
1.1 Antecedentes	2
1.2 Planteamiento del problema	2
1.3 Alcance	3
1.4 Justificación del tema	4
1.5 Objetivos	5
1.5.1 Objetivo general	5
1.5.2 Objetivos específicos	5
1.6 Marco Metodológico	5
1.6.1 Análisis y selección de la metodología	5
1.6.2 Metodología Rup.	6
1.6.3 Metodología Xp.	7
1.6.4 Comparación de las metodologías Rup, Xp.	9
1.6.5 Selección de la metodología Rup.	10
<b>CAPÍTULO 2</b>	<b>12</b>
<b>MARCO TEÓRICO</b>	<b>12</b>
2.1 Portales web.	12
2.2 Sistema Gestores de base de datos	14
2.3 Visual Studio C# (C Sharp).	16
<b>CAPÍTULO 3</b>	<b>18</b>
<b>ANÁLISIS Y DISEÑO</b>	<b>18</b>
3.1 Análisis de la situación actual	18
3.2 Planteamiento de mejora de procesos.	19
3.3 Diseño del sistema	21
3.3.1 Diagrama lógico de la base de datos	21
3.3.2 Diagrama físico de la base de datos	21
3.3.3 Diagrama de clases	29
3.3.4 Casos de uso	30
3.3.5 Diagrama de secuencia	34

3.3.6 Diagrama de estado	38
3.3.7 Diagrama navegacional	42
3.3.8 Glosario	46
3.4 Aplicación de la metodología Rup en el desarrollo del software	47
3.4.1 Fase de inicio.	47
3.4.2 Fase de elaboración	52
<b>CAPÍTULO 4</b>	<b>54</b>
<b>CONSTRUCCIÓN Y PRUEBAS</b>	<b>54</b>
4.1 Fase Desarrollo	54
4.1.1 Interfaz de usuario	54
4.1.2 Codificación	58
4.2 Pruebas	64
4.3 Fase de transición	69
4.3.1 Manual de instalación.	69
4.3.2 Manual de Usuario.	69
<b>CONCLUSIONES</b>	<b>70</b>
<b>RECOMENDACIONES</b>	<b>71</b>
<b>ANEXOS</b>	<b>74</b>

## ÍNDICE DE TABLAS

Tabla 1. Principios importantes de Rup.	7
Tabla 2. Valores de la programación Xp.	8
Tabla 3. Comparación entre Metodologías RUP y XP.	9
Tabla 4. Glosario.	46
Tabla 5. Prueba caja negra ingreso usuario	65
Tabla 6. Control de prueba ingreso usuario	66
Tabla 7. Prueba caja negra matricula usuario	67
Tabla 8. Control de prueba matricula usuario	68

## ÍNDICE DE FIGURAS

Figura 1. Diagrama de modelo lógico. _____	22
Figura 2. Diagrama de modelo físico _____	23
Figura 3. Diagrama de clases _____	29
Figura 4. Modelo de caso de uso registro estudiante. _____	30
Figura 5. Modelo de caso de uso registro estudiante. _____	31
Figura 6. Modelo de caso de uso asignación horario. _____	32
Figura 7. Modelo de caso de uso asignación capacitador. _____	33
Figura 8. Diagrama de secuencia para registro de estudiante. _____	34
Figura 9. Diagrama de secuencia para matriculación de estudiante. _____	35
Figura 10. Diagrama de secuencia para asignación horario. _____	36
Figura 11. Diagrama de secuencia para asignación capacitador. _____	37
Figura 12. Diagrama de estado registro estudiante. _____	38
Figura 13. Diagrama de estado matriculación de estudiante. _____	39
Figura 14. Diagrama de estado asignación horario. _____	40
Figura 15. Diagrama de estado asignación de capacitador. _____	41
Figura 16. Diagrama navegacional registro estudiante. _____	42
Figura 17. Diagrama navegacional matriculación de estudiante. _____	43
Figura 18. Diagrama navegacional asignación de horario. _____	44
Figura 19. Diagrama navegacional asignación de capacitador. _____	45
Figura 20. Secretaria modelo de caso de uso. _____	50
Figura 21. Facilitador modelo de caso de uso. _____	50
Figura 22. Alumno modelo de caso de uso. _____	51
Figura 23. Modelamiento de la base de datos.. _____	52
Figura 24. Diagrama conceptual entidad - relación _____	53
Figura 25. Interfaz de usuario inicio de sesión. _____	55

Figura 26. Interfaz de usuario estándar	56
Figura 27. Interfaz de usuario registro de datos.	57
Figura 28. Script de la base de datos.	58

## ÍNDICE DE ANEXOS

Anexo 1. Manual de usuario	74
Anexo 2. Manual técnico	80

## **RESUMEN**

Este proyecto técnico se ha desarrollado para servir en el proceso de: la inscripción de los estudiantes, la matrícula en los talleres y la asignación de capacitadores, entre otros. Esto provoca retrasos en la ejecución de la entrega de diplomas, certificados, trámites de legalización y que afecta a la toma de decisiones, lo que resulta en la pérdida de información y tiempo para los usuarios que necesitan información precisa, disponible, oportuna y confiable.

El software tiene como alcance el investigar, analizar y proponer una solución para crear un sistema de ayuda social, con el fin de que se cumplan todas las expectativas que la Federación de barrios del noroccidente de Quito anhela. El software es desarrollado en ambiente web.

## **ABSTRACT**

This technical project is developed to serve in the process: registration of students, enrollment in workshops and assignment of teachers, among others. This causes delays in implementation for the delivery of diplomas, certifications, legalization procedures and affecting decision-making, resulting in loss of information and time to users who need accurate, available, timely and reliable information.

The software is scoped to investigate, analyze and propose the best solution to create a welfare system, so that all expectations Federation suburbs northwest of Quito longs are met. The software is developed in web environment.

## **INTRODUCCIÓN**

La Federación de Barrios del Noroccidente de Quito es una organización que tiene como finalidad ofrecer cursos, talleres de capacitación sobre temas relacionados con tecnología, gastronomía y finanzas entre otros; dirigido a los niños, adolescentes y adultos mayores en el desarrollo de valores, responsabilidades, sin descuidar el aspecto social y cultural, lo que permite brindar un servicio eficiente y ágil.

En el capítulo I se identifica la problemática a tratar, que será la base y permitirá dar a conocer el propósito de desarrollo a este trabajo, así como los objetivos que se pretende alcanzar, los lineamientos que se desea conocer y una descripción general del software a desarrollar.

En el capítulo II se obtiene la definición de la arquitectura del sistema y una especificación detallada de los componentes del sistema de información.

En el capítulo III y IV se realiza el análisis, diseño, construcción y pruebas de operación y rendimiento del sistema, dichas pruebas son realizadas para la verificación del buen estado y funcionamiento del mismo.

# CAPÍTULO 1

## 1.1 Antecedentes

El software informático en general es de gran ayuda en las tareas que desempeña la Federación de Barrios del Noroccidente de Quito al momento de realizar:

- El registro de los estudiantes nuevos.
- El registro de facilitadores, cursos y horarios nuevos.
- El registro de módulos nuevos dependiendo el curso.
- El registro de año lectivo.
- La matriculación de los estudiantes en cursos disponibles.
- La asignación de facilitadores disponibles para los cursos y horarios ofertados.
- La asignación de horarios para los cursos dependiendo de la disponibilidad del curso y facilitador.

## 1.2 Planteamiento del problema

La Federación de Barrios del Noroccidente de Quito es una organización cuya finalidad es ofrecer capacitaciones a los niños, adolescentes y adultos mayores en el desarrollo de valores, responsabilidades, sin descuidar el aspecto social y cultural.

El proceso de registro y matriculación se los realiza en forma manual. Esto produce retardos en la ejecución para la entrega de diplomas, certificaciones, legalización de trámites, afectando la toma de decisiones, ocasionando pérdida de información y tiempo en los usuarios (moradores de los barrios, docentes, directivos y niños) que necesitan información exacta, disponible, oportuna y confiable.

Considerando los inconvenientes mencionados, se presenta la necesidad de diseñar un sistema que permita gestionar los procesos indicados anteriormente.

### **1.3 Alcance**

El software tiene como alcance el investigar, analizar y proponer una solución para crear un sistema de ayuda social, con el fin de que se cumplan todas las expectativas que la Federación de barrios del noroccidente de Quito anhela. El software es desarrollado en ambiente web y se organiza en los siguientes módulos:

#### ***Proceso de Administración de parámetros***

Con este módulo se identificará la información necesaria que permita definir la funcionalidad del sistema y fijará parámetros a utilizar en los procesos de: registro de alumnos, matriculación de talleres, toma de horarios y asignación de facilitador a los talleres.

#### ***Proceso de matriculación a los diferentes talleres***

Con este módulo se cubrirá el objetivo de automatizar los procesos de matriculación a los talleres de capacitación, verificando la disponibilidad de los talleres y el cupo asignado.

### ***Proceso de control para la generación de horarios***

Con este módulo se cubrirá el objetivo de controlar los horarios de los talleres de capacitación, la asignación de facilitador en el periodo presente.

### ***Proceso de reportes***

Con este módulo se cubrirá el objetivo de la generación de reportes como son:

- Reporte de facilitadores.
- Reporte estudiantes matriculados.
- Reporte de talleres.

### **1.4 Justificación del tema**

El análisis y desarrollo del software informático que gestiona los talleres de capacitación será un gran aporte para cubrir las necesidades de registro de estudiantes, creación de talleres, facilitadores, horarios y cupo disponibles, generación de reportes utilizados para llevar el control y estadística, asignación de talleres, horarios, año lectivo y estudiantes verificando la disponibilidad que tiene la Federación de Barrios del Noroccidente de Quito.

Esto va a permitir que la Federación de Barrios del Noroccidente de Quito brinde un servicio eficiente y ágil, donde el personal administrativo contará con información completa y oportuna referente a los talleres de capacitación, los estudiantes registrados por talleres y los facilitadores asignados, evitando de esta manera que los

procesos sean llevados manualmente ocasionando perdida de datos, lentitud en el proceso y falta de información a tiempo.

## **1.5 Objetivos**

### **1.5.1 Objetivo general**

Analizar, diseñar y desarrollar un Sistema Informático que gestione los talleres de capacitación organizados por la Administración de la Federación de Barrios del Noroccidente de Quito.

### **1.5.2 Objetivos específicos**

- Realizar el estudio sobre las metodologías para desarrollo de proyectos o de software y determinar la más adecuada para la realización de proyecto propuesto.
- Realizar el análisis correspondiente para la asignación de horarios en función de la disponibilidad de los facilitadores y los horarios asignados a temáticas de los temas a ser tratados.
- Diseñar y construir el sistema informático en ambiente web que gestione el proceso de inscripción, matriculación a los talleres de capacitación organizados por la Administración de la Federación de Barrios del Noroccidente de Quito.

## **1.6 Marco Metodológico**

### **1.6.1 Análisis y selección de la metodología**

En este punto se realiza la revisión de dos metodologías que se tomaron en consideración para ser utilizadas en el proceso de desarrollo de software y se analizaran en una breve descripción. De las metodologías tomadas en cuenta tenemos que destacar las características, ventajas y desventajas; sin embargo, la selección y utilización de una de ellas dependerá de los requerimientos y necesidades del proyecto.

### **1.6.2 Metodología Rup**

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones. RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. (Wikispaces, 2012)

## Principios de desarrollo Rup

Tabla 1.

Principios importantes de Rup.

<b>Principios</b>	<b>Descripción</b>
<b>Adaptar el proceso</b>	El proceso debe adaptarse a las características de la organización para la que se está desarrollando el software. (Wikispaces, 2012)
<b>Equilibrar prioridades</b>	Debe encontrarse un balance que satisfaga a todos los inversores del proyecto. (Wikispaces, 2012)
<b>Demostrar valor iterativamente</b>	Los proyectos se entregan, aunque sea de una forma interna, en etapas iteradas. (Wikispaces, 2012)
<b>Colaboración entre equipos</b>	Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, entre otros. (Wikispaces, 2012)
<b>Enfocarse en la calidad</b>	La calidad del producto debe verificarse en cada aspecto de la producción. (Wikispaces, 2012)

Nota: Principios de desarrollo de Rup.

Elaborado por: Renato Díaz y Jorge Carua

### 1.6.3 Metodología Xp

La filosofía de X.P es satisfacer al completo las necesidades del cliente, por eso lo integra como una parte más del equipo de desarrollo. Xp fue inicialmente creada para el desarrollo de aplicaciones dónde el cliente no sabe muy bien lo que quiere, lo que provoca un cambio constante en los requisitos que debe cumplir la aplicación. Por este motivo es necesaria una metodología ágil como Xp que se adapta a las

necesidades del cliente y dónde la aplicación se va reevaluando en periodos cortos de tiempo. Xp está diseñada para el desarrollo de aplicaciones que requieran un grupo de programadores pequeño, dónde la comunicación sea más factible que en grupos de desarrollo grandes. La comunicación es un punto importante y debe realizarse entre los programadores, los jefes de proyecto y los clientes. (Lycos, 1994)

### Valores de la programación extrema Xp

Tabla 2.

Valores de la programación Xp.

<b>Valores</b>	<b>Descripción</b>
<b>Simplicidad</b>	Se comienza desarrollando las soluciones más sencillas necesarias para solucionar los problemas. (Pmoinformatica, 2012)
<b>Comunicación</b>	La comunicación se realiza por medio de transferencia de conocimientos en reuniones frecuentes cara a cara entre usuarios y desarrolladores. (Pmoinformatica, 2012)
<b>Realimentación</b>	Mostraremos al usuario nuestro software frecuentemente y de forma temprana, escuchando cuidadosamente sus observaciones y realizando los cambios que sean necesarios. (Pmoinformatica, 2012)
<b>Coraje o Valentía</b>	Nos adaptaremos a los cambios cuando sea que estos ocurran. (Pmoinformatica, 2012)
<b>Respeto</b>	Es tanto por el trabajo de los demás como por el trabajo de uno mismo. (Pmoinformatica, 2012)

Nota: Valores de la programación extrema Xp.

Elaborado por: Renato Díaz y Jorge Carua

#### 1.6.4 Comparación de las metodologías Rup, Xp

En este punto vamos a detallar las características más relevantes de las metodologías Rup y Xp que fueron tomadas en cuenta para identificar la mejor opción con respecto a los procesos para el desarrollo del software.

Las dos metodologías se enfocan en el desarrollo ordenado de los procesos, tomando en consideración un esquema sucesivo manteniendo prácticas que soportan el desarrollo y el ciclo de la calidad del proyecto. De acuerdo a nuestra investigación concluimos que estas metodologías nos brindan disponibilidad en los procesos de tal forma que podemos interactuar con la documentación, realizar cambios durante el desarrollo para obtener un software final que cumpla con los requisitos que se proponen en el proyecto.

El siguiente cuadro comparativo de las metodologías Rup y Xp servirá de apoyo para enfocarnos en la selección del proceso más indicado para el desarrollo del software, ver tabla 3.

Tabla 3.

Comparación entre Metodologías RUP y XP.

<b>Descripción</b>	<b>RUP</b>	<b>XP</b>
Desarrollo de software iterativo	<b>X</b>	<b>X</b>
Procesos de calidad como factor importante en el diseño	<b>X</b>	
Requerimientos del cliente	<b>X</b>	<b>X</b>
Procesos en equipo	<b>X</b>	<b>X</b>

Restricciones en procesos	X	
Gestión de cambios y configuraciones	X	
Validación de la calidad en la continuidad de los procesos	X	X

Nota: Comparación de metodologías.

Elaborado por: Renato Díaz y Jorge Carua

### **1.6.5 Selección de la metodología Rup**

Tomando en consideración el estudio que se realizó anteriormente de las metodologías Rup y Xp, determinamos que Rup es la mejor opción para el desarrollo del software, nos brinda los mejores procesos para el ciclo de vida del proyecto y se describen a continuación.

#### ***Fase de Inicio***

Esta fase tiene como propósito definir y acordar el alcance del proyecto con los usuarios, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software, producir el plan de las fases y el de iteraciones posteriores. (Wikispaces, 2012)

#### ***Fase de elaboración***

En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema, el desarrollo, las especificaciones, el primer análisis del problema y se diseña la solución preliminar. (Wikispaces, 2012)

### ***Fase de Desarrollo***

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto. (Wikispaces, 2012)

### ***Fase de Cierre***

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto. (Wikispaces, 2012)

### **Características**

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo). (Wikispaces, 2012)
- Pretende implementar las mejores prácticas en Ingeniería de Software. (Wikispaces, 2012)
- Desarrollo iterativo. (Wikispaces, 2012)
- Administración de requisitos. (Wikispaces, 2012)
- Uso de arquitectura basada en componentes. (Wikispaces, 2012)
- Control de cambios. (Wikispaces, 2012)
- Modelado visual del software. (Wikispaces, 2012)
- Verificación de la calidad del software. (Wikispaces, 2012)

## **CAPÍTULO 2**

### **MARCO TEÓRICO**

#### **2.1 Portales web**

Un Portal web es el punto de entrada a Internet donde se organizan y concentran los contenidos del propietario del mismo, el objetivo principal del Portal es ayudar a los usuarios a encontrar lo que necesitan sin salir del mismo e incentivándoles a utilizarlo de forma continuada. Los 3 pilares fundamentales de un Portal para atraer la atención del usuario son: Información (Buscadores, directorios, noticias, catálogos y servicios), Participación (E-mail, foros, chat), Comodidad (brindar la mayor cantidad de información en un solo espacio y tenerlo todo a mano). (Buyto, 2009)

#### **Tipos de portales según sus características**

**Portales Horizontales:** Es un portal con contenidos de carácter general orientado a todo tipo de usuario. Incluye servicios tal cómo: Comunidades virtuales, Chat, e-mail, Espacio web gratuito, Foros de discusión, Noticias, Información, Buscadores, etc. Son portales para usuarios no muy experimentados en el mundo de Internet a los que se les brinda una gran gama de información y servicios sin obligarles a navegar por varios sitios para obtenerlos. Como ejemplo de portales de esta categoría están: Terra, AOL, AltaVista, UOL, Lycos, Yahoo, MSN. (Buyto, 2009)

**Portales Verticales:** Es un sitio web que provee de información y servicios a un sector en particular, con contenidos concretos y centrados en un tema como puede ser un portal de música, empleo, inmobiliario, un portal de finanzas, arte o deportes. (Buyto, 2009)

Portales Diagonales: Se trata de una mezcla entre el portal horizontal y el vertical. Utilizan redes sociales o aplicaciones generalistas como Facebook, LinkedIn, Flickr o YouTube, complementadas con contenidos y/o utilidades dirigidas a un público muy concreto. (Buyto, 2009)

Portales Especializados: Surgen a partir de la demanda de los usuarios de un sitio más especializado en sus áreas de interés tanto particulares como profesionales y que los Portales horizontales y verticales no son capaces de llenar por tener contenidos demasiados generales o superficiales. Hoy en día existen infinidad de Portales exclusivos en temas específicos como, educación, zonas geográficas, aficiones, deportes, conocimiento, etc. (Buyto, 2009)

Portales Corporativos: Son una prolongación de la intranet de las empresas, donde se potencia el acceso a la información de la organización, contacto con clientes y proveedores, material de trabajo a sus trabajadores, cuidando mucho el tipo de información a presentar en función del tipo de usuario que la solicita por ser muchas veces confidencial. Incluyen ayudas internas para buscar documentación, personalizar el portal para diferentes grupos de usuarios, etc. (Buyto, 2009)

Portales Móviles: Son sitios que permiten la conexión de los usuarios tanto desde Internet como de un teléfono móvil o PDA. Dichos portales se están abriendo camino a nivel de empresas para realizar negocios ofreciendo productos o servicios a los usuarios. En la actualidad no existen muchos portales móviles, pero con la cantidad de móviles existentes será uno de los modelos que crezca en la red durante estos años. (Buyto, 2009)

## 2.2 Sistema Gestores de base de datos

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos. (Ecuared, 2004)

A continuación ponemos a consideración varios sistemas de gestor de base de datos.

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes. Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras. (Ecuared, 2004)

SQL Server es un sistema gestor de base de datos relacionales producido por Microsoft. Es un sistema cliente/servidor que funciona como una extensión natural del sistema operativo Windows. Entre otras características proporciona integridad de datos, optimización de consultas, control de concurrencia, backup y recuperación. Es relativamente fácil de administrar a través de la utilización de un entorno gráfico para casi todas las tareas de sistema y administración de bases de datos. Utiliza servicios del sistema operativo Windows para ofrecer nuevas capacidades o ampliar la base de datos, tales como enviar y recibir mensajes y gestionar la seguridad de la conexión. Es fácil de usar y proporciona funciones de almacenamiento de datos que sólo estaban disponibles en Oracle y otros sistemas gestores de bases de datos más caros. (Ecuared, 2004)

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos, derivado de Postgres, desarrollado en la Universidad de California, en el Departamento de Ciencias de la Computación de Berkeley. Es un gestor de bases de datos de código abierto, brinda un control de concurrencia multi-versión (MVCC por sus siglas en inglés) que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación. Posee características significativas del motor de datos, entre las que se pueden incluir las subconsultas, los valores por defecto, las restricciones a valores en los campos (constraints) y los disparadores (triggers). Ofrece funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales. (Ecuared, 2004)

Dentro de los sistemas gestores de base de datos se escogió PostgreSQL, ya que dentro de las características que ofrece se puede identificar funciones destacadas para el desarrollo de la base de datos. PostgreSQL al ser un sistema orientado a objetos y al manejar multiprocesos garantiza la estabilidad del sistema.

PostgreSQL tiene las siguientes características en las cuales nos enfocamos para brindar la disponibilidad con la información que maneja la Federación:

- Aislamiento: Asegura que una operación no puede afectar a otras.
- Durabilidad: Durabilidad de procesos así falle el sistema.
- Multiplataforma: Compatible con: (Linux, unix, Windows, etc).
- Adaptable: Factible a requerimientos solicitados por el cliente.
- Soporte: Soporta tipo de datos definidos por el usuario.

### **2.3 Visual Studio C# (C Sharp)**

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Ésta es la llamada plataforma .NET, y a los servicios antes comentados se les denomina servicios Web. (González, 2016, p. 9)

C# “C Sharp” es el nuevo lenguaje diseñado por Microsoft para su plataforma .NET, aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET. La sintaxis y estructuración de C# es muy similar a la C++, ya que la intención de Microsoft con C# es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin

embargo, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic. (González, 2016, p. 21)

C# es la herramienta con la que desarrollamos el software y nos brinda de gran ayuda su lenguaje de programación, el cual desarrolla las mejoras que necesitamos con respecto a las características de los lenguajes existentes como Visual Basic, Java o C++ y las combina en uno solo, haciendo de C# una plataforma compacta y estable para lo que necesita la Federación. A partir de las siguientes ventajas podemos determinar que C# es la mejor opción para el desarrollo del software.

- Sencillez: C# maneja elementos que otros lenguajes incluyen y que son innecesarios en .NET.
- Modernidad: C# incorpora nuevos elementos en el lenguaje útiles para el desarrollo de aplicaciones.
- Orientación a objetos: C# reduce problemas por conflictos de nombres y suministra la legibilidad del código.

## **CAPÍTULO 3**

### **ANÁLISIS Y DISEÑO**

#### **3.1 Análisis de la situación actual**

La Federación de Barrios del Noroccidente de Quito ofrece talleres de capacitación orientada a los niños, adolescentes y adultos mayores de los barrios que actualmente se encuentran dentro de la gestión. Estos procesos se documentan de forma manual, la inscripción a los diferentes talleres además de la generación de horarios y asignación de capacitadores que maneja la Federación.

#### **Levantamiento de información**

Para determinar en forma detallada el estado actual de los procesos que la Federación lleva a cabo de forma manual se tomará en cuenta los siguientes: registro, matriculación, asignación de horarios y capacitador.

Registro de estudiantes: Este proceso se lleva a cabo cuando el estudiante ingresa por primera vez a formar parte de los registros de la Federación.

Matriculación de estudiante: Este proceso se lleva a cabo cuando el estudiante está interesado en formar parte de la capacitación que la Federación ofrece con su respectivo taller.

Asignación de horarios: Este proceso se realiza una vez que el estudiante ya está registrado a un taller y desea tomar un horario de acuerdo a la disponibilidad.

Asignación de capacitador: Este proceso se realiza cuando el taller esta creado.

### **3.2 Planteamiento de mejora de procesos**

En este punto se enumerará las mejoras de acuerdo a la situación actual donde podemos verificar los procesos manuales que la Federación lleva a cabo y se automatizarán de acuerdo a la información obtenida. Para ello tomaremos los procesos antes ya descritos y detallaremos las mejoras de acuerdo a la necesidad.

Registro de estudiantes: Este proceso cuenta con las siguientes mejoras:

- Plantilla de registro amigable y de fácil uso
- Registro correcto de la información
- Disponibilidad inmediata de la información
- Información completa del estudiante siguiendo un estándar definido
- Manejo de reporte de estudiantes

Matriculación de estudiante: Este proceso se llevará a cabo una vez que el estudiante este registrado en la base de datos y cuente con la información necesaria para ser merecedor del registro a un taller según la disponibilidad. Las mejoras de este proceso son:

- Información oportuna del estudiante
- Información detallada de los talleres disponibles
- Verificación y control de los talleres para evitar cruces
- Módulos disponibles al tener un taller ya asignado
- Manejo de reporte de estudiantes matriculados por taller

Asignación de horarios: Este proceso se lleva a cabo un vez que el estudiante este asignado y debidamente verificado la disponibilidad del taller. Las mejoras de este proceso son:

- Información oportuna de los horarios disponibles para el taller correspondiente
- Verificación y control de los horarios disponibles
- Registro correcto de la asignación de horarios
- Control de las horas por día para evitar posibles errores

Asignación de capacitador: Este proceso se lleva a cabo una vez que el taller se encuentre creado con sus respectivos módulos y año lectivo en curso. Las mejores de este proceso son:

- Información oportuna de los talleres disponibles, los módulos asignados y del año lectivo en curso
- Verificación y control de la disponibilidad del capacitador
- Control se asignación de talleres de acuerdo a la disponibilidad del capacitador
- Manejo de reportes de capacitadores disponibles

### **3.3 Diseño del sistema**

#### **3.3.1 Diagrama lógico de la base de datos**

Una vez realizada la etapa de análisis seguiremos con la de diseño del sistema, donde pasamos al diseño lógico el cual permite estructurar el flujo y la relación de la base de datos, donde cada entidad será una tabla y cada atributo un campo de la tabla. Y el atributo identificador será la clave primaria de la tabla. Además pueden aparecer tablas extras que son necesarias para representar algunos tipos de relaciones, ver Figura 1.

#### **3.3.2 Diagrama físico de la base de datos**

Se obtiene a partir del modelo lógico, optimiza el rendimiento a la vez que asegura la integridad de los datos al evitar repeticiones innecesarias de datos. Durante el diseño físico, se transforman las entidades en tablas, las instancias en filas y los atributos en columnas. Este diseño se adaptará la base de datos, ver Figura 2.

Diagrama de Modelo Lógico

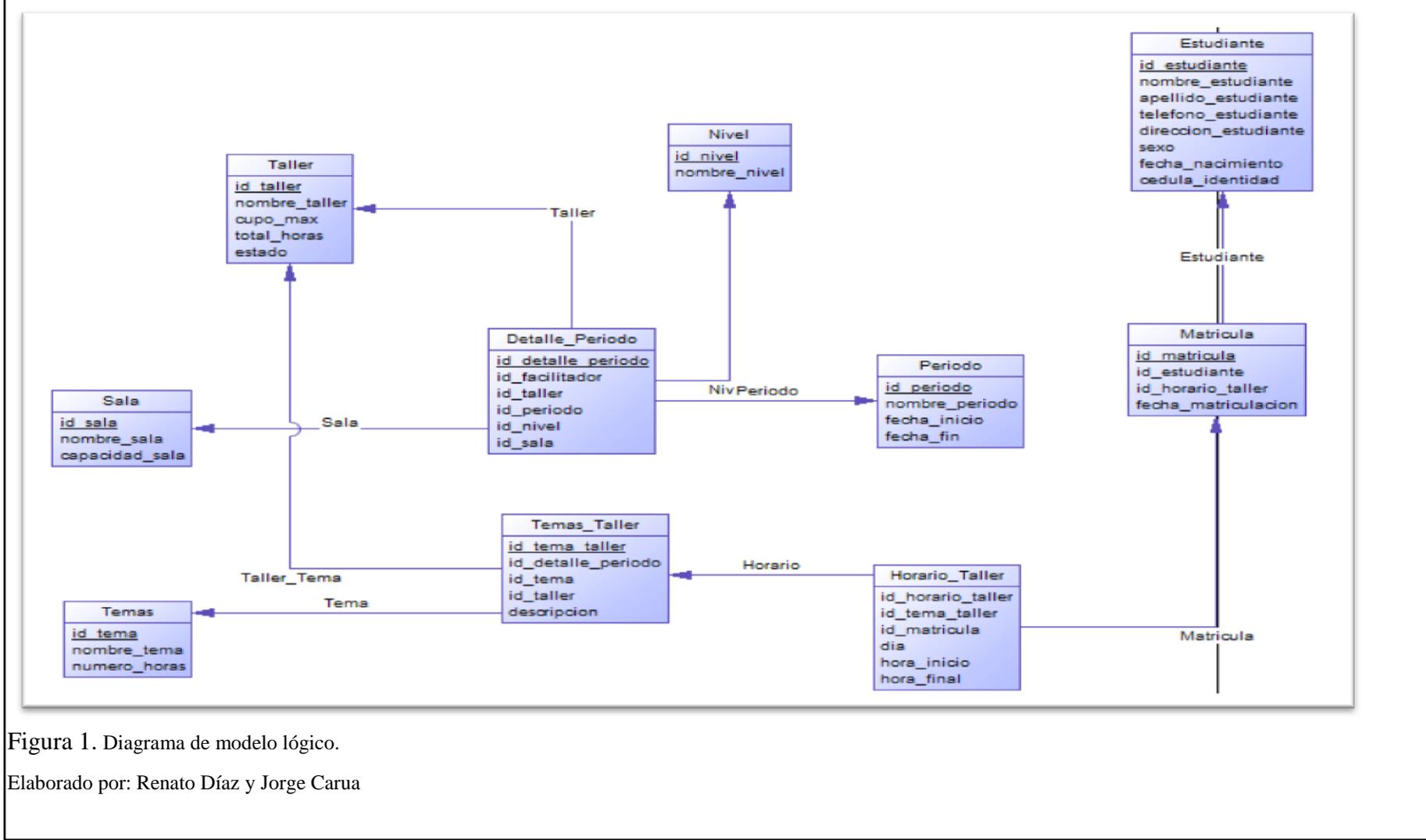


Figura 1. Diagrama de modelo lógico.

Elaborado por: Renato Díaz y Jorge Carua

## Diagrama de Modelo Físico

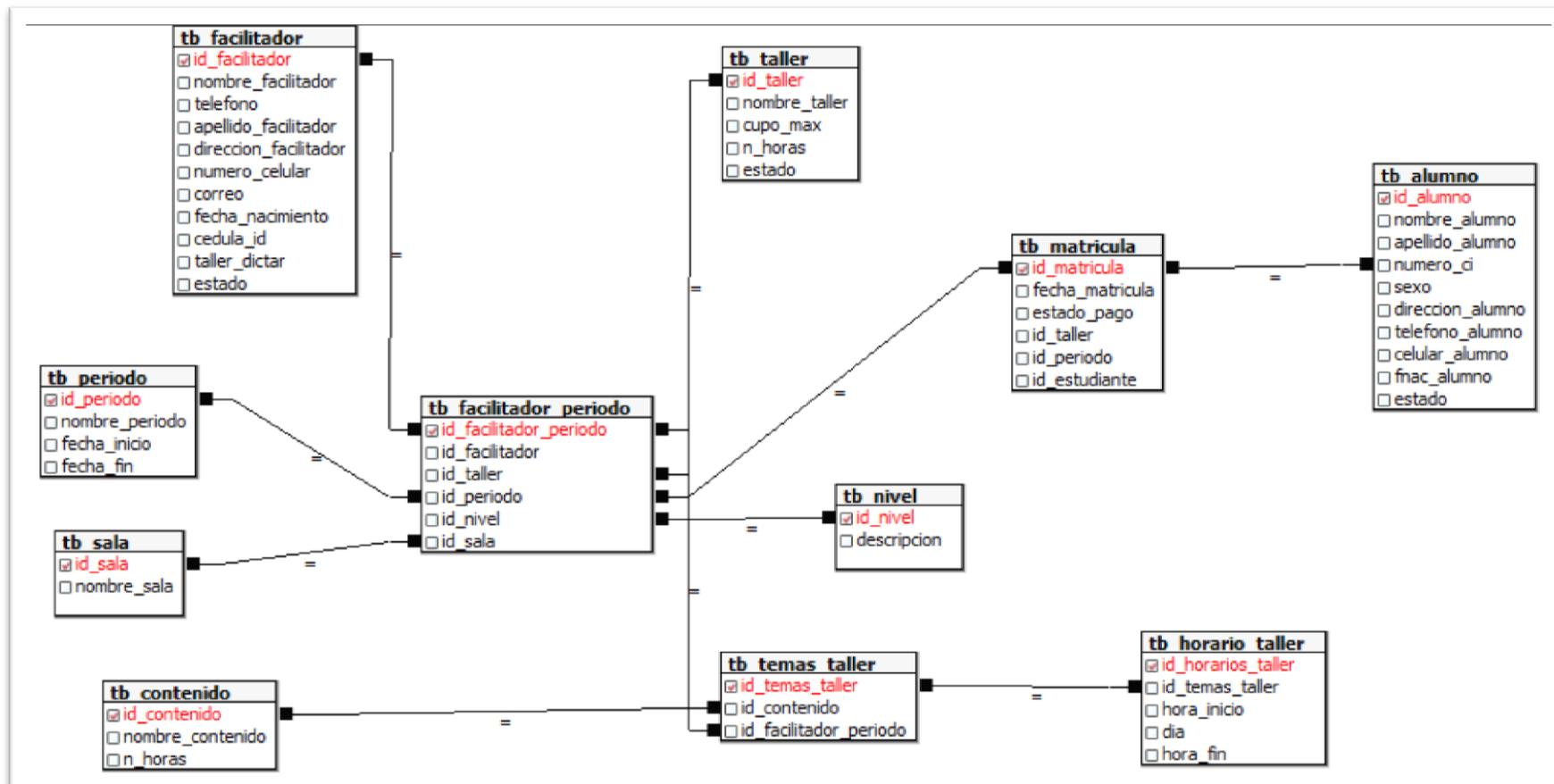


Figura 2. Diagrama de modelo físico

Elaborado por Renato Díaz y Jorge Carua

## Descripción de Tablas

### *Lista de Tablas.*

#### Tabla tb\_facilitador

Descripción: Se almacena la información de personas que van hacer encargadas de dictar los talleres.

Clave primaria, foráneas e índices.

Nombre	Clave Primaria	Clave Foránea
tb_facilitador_pkey	X	

#### Tabla tb\_alumno

Descripción: Se almacena la información de personas inscritas en la Federación.

Clave primaria, foráneas e índices.

Nombre	Clave Primaria	Clave Foránea
id_alumno_pkey	X	

#### Tabla tb\_taller

Descripción: Se almacena la información de los talleres que se dictarán en los diferentes niveles durante el presente periodo.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_taller_pkey	X	

Tabla tb\_periodo

Descripción: Se almacena la información de los periodos que se crearán en el año.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_periodo_pkey	X	

Tabla tb\_nivel

Descripción: Se almacena la información de los niveles creados para un determinado taller y periodo.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_nivel_pkey	X	

Tabla tb\_sala

Descripción: Se almacena las salas que se encuentran disponibles para que dicten los diferentes talleres.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_sala_pkey	X	

Tabla tb\_contenido

Descripción: Se almacena el contenido que se dictara en los diferentes niveles durante el presente periodo.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_contenido_pkey	X	

Tabla tb\_matricula

Descripción: Se almacena la información de las personas que se encuentran matriculados en la Institución.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_matricula_pkey	X	
estudiante_matricula_fk		X
facilitador_matricula_fk		x

Tabla tb\_menu

Descripción: Se almacena la información de relación que existe entre un perfil y un usuario para que pueda acceder al sistema dependiendo del perfil asignado.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_menu_pkey	X	

Tabla tb\_perfil

Descripción: Se almacena la información de los perfiles que pueden tener los usuarios.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_perfil_pkey	X	

Tabla tb\_menuperfil

Descripción: Se almacena la información de relación que se tiene entre un perfil y un módulo del sistema el cual fue asignó a un perfil

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_menu_perfil_pkey	X	
id_menu_fk		X
id_perfil_fk		X

Tabla tb\_facilitadorperido

Descripción: Se almacena la información de las personas que se encuentran asignadas a un determinado taller, nivel para el periodo.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_facilitador_periodo_pkey	X	
facilitador_fk		X
nivel_fk		X
periodo_fk		X
sala_fk		X
taller_fk		X

Tabla tb\_ temastaller

Descripción: Se almacena la información de los contenidos asignados a los talleres.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_temas_taller_pkey	X	
contenido_fk		X
facilitador_periodo_fk		X

Tabla tb\_ horariostaller

Descripción: Se almacena la información de los talleres en días y horas de la semana.

Clave primaria, foráneas.

Nombre	Clave Primaria	Clave Foránea
id_horarios_taller_pkey	X	
temas_taller_fk		X



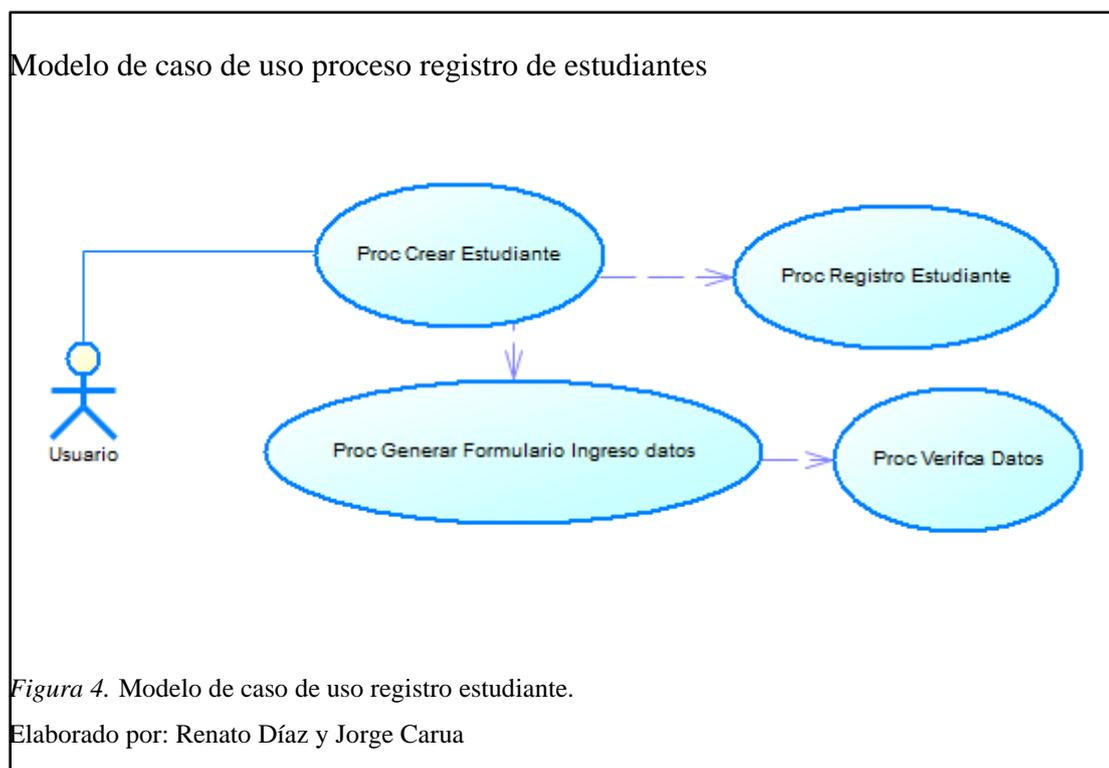
### 3.3.4 Casos de uso

#### Modelo de caso de uso

A continuación se presenta los diagramas de casos de uso determinados para el diseño del proyecto por varios procesos de anteriormente se habían verificado su automatización los cuales son: Registro de estudiantes al sistema, asignación de taller previa validación de disponibilidad, asignación de horario previa verificación de taller y asignación de capacitador previa validación de disponibilidad por tema de horarios.

#### Registro de estudiantes

Este diagrama de caso de uso nos muestra la gestión que realizada al registrar un estudiante nuevo.

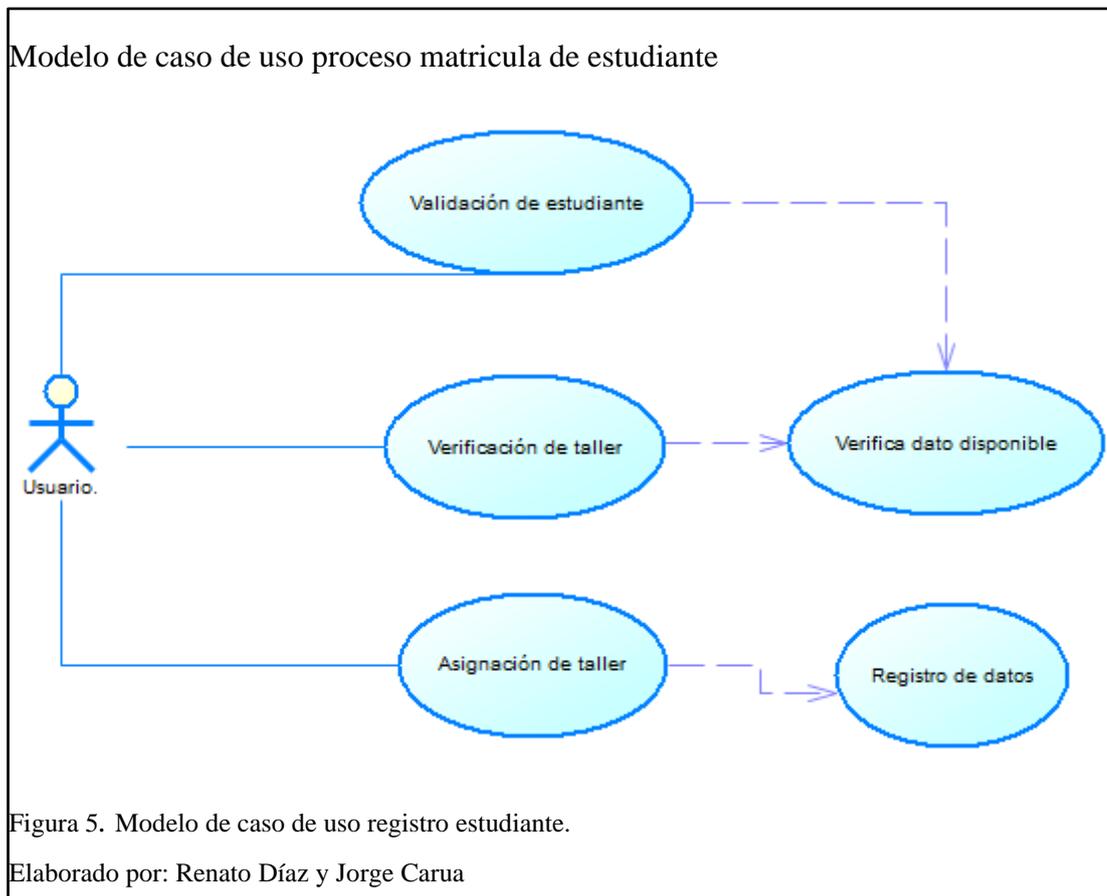


Flujo de eventos

El usuario crea un nuevo estudiante y registra los datos correspondientes.

### Matriculación estudiante

Este diagrama de caso de uso nos muestra la gestión realizada por un usuario al matricular un nuevo estudiante en un taller disponible, verificando que exista en el registro de talleres.

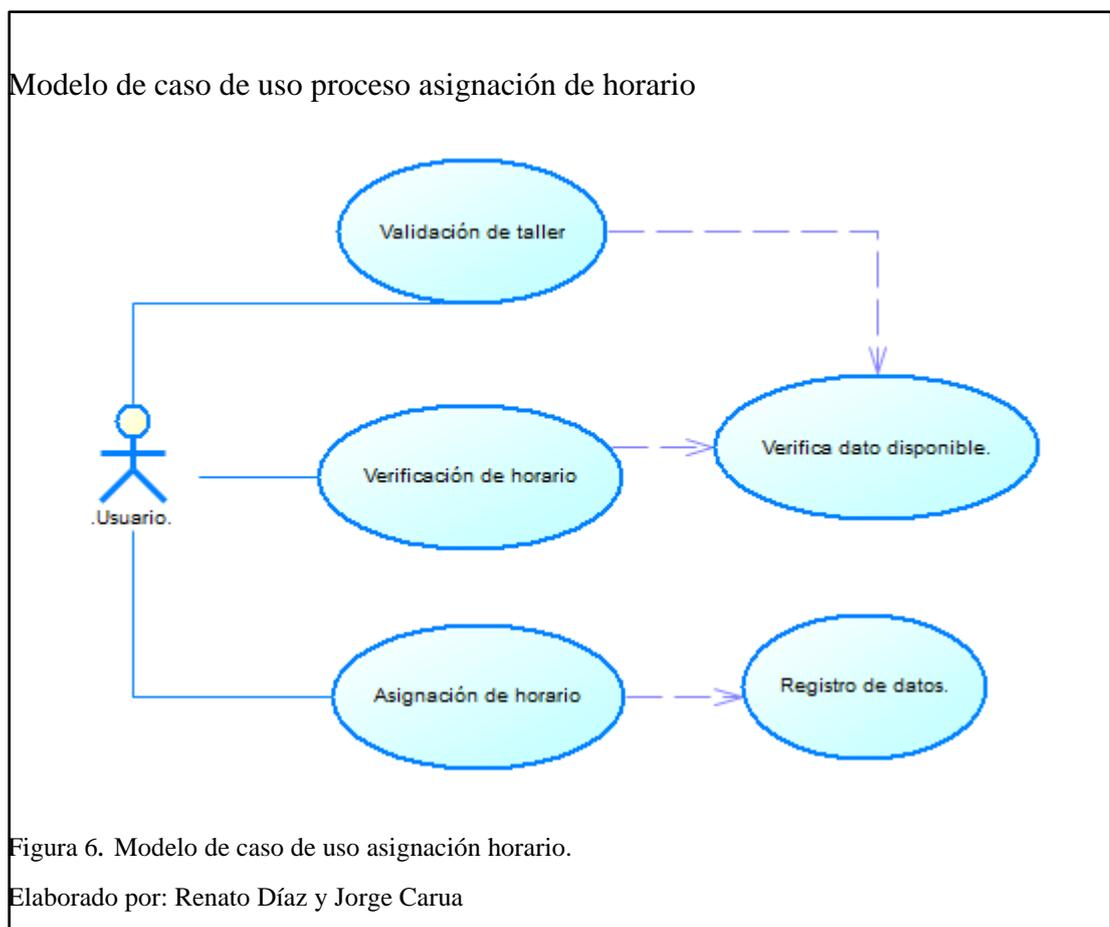


Flujo de eventos

El usuario verifica si el estudiante se encuentra en los registros, si el taller a escoger está disponible y de ser el caso asigna al estudiante el taller.

## Asignación de horarios

Este diagrama de caso de uso nos muestra la gestión realizada al asignar un horario al taller que el estudiante solicito previa validación de disponibilidad y verificación de cruces no deseados.

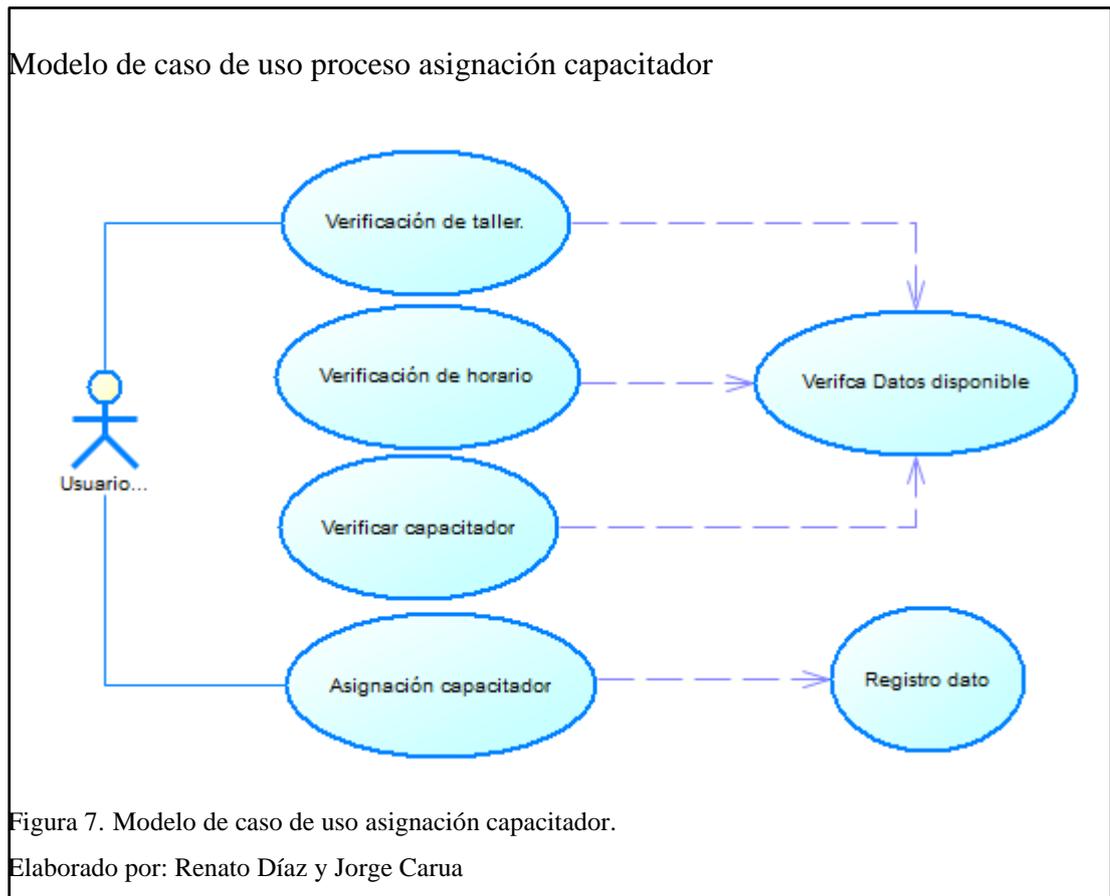


### Flujo de eventos

El usuario verifica si el taller tiene horarios disponibles y los asigna al taller que el estudiante desee registrarse.

## Asignación de capacitador

Este diagrama de caso de uso nos muestra la gestión realizada al asignar un capacitador a los talleres disponibles y previa verificación del año lectivo en curso para evitar registros incorrectos.



### Flujo de eventos

El usuario verifica si el taller tiene horarios disponibles, si el capacitador está creado y asigna al capacitador al taller correspondiente validando los parámetros establecidos de tiempo.

### 3.3.5 Diagrama de secuencia

Diagrama de secuencia utilizado para el diseño del software tomando en cuenta los procesos automáticos antes ya descritos en la fase análisis.

#### Diagrama de secuencia registro estudiante

Este diagrama muestra el proceso de registro de un estudiante nuevo al sistema.

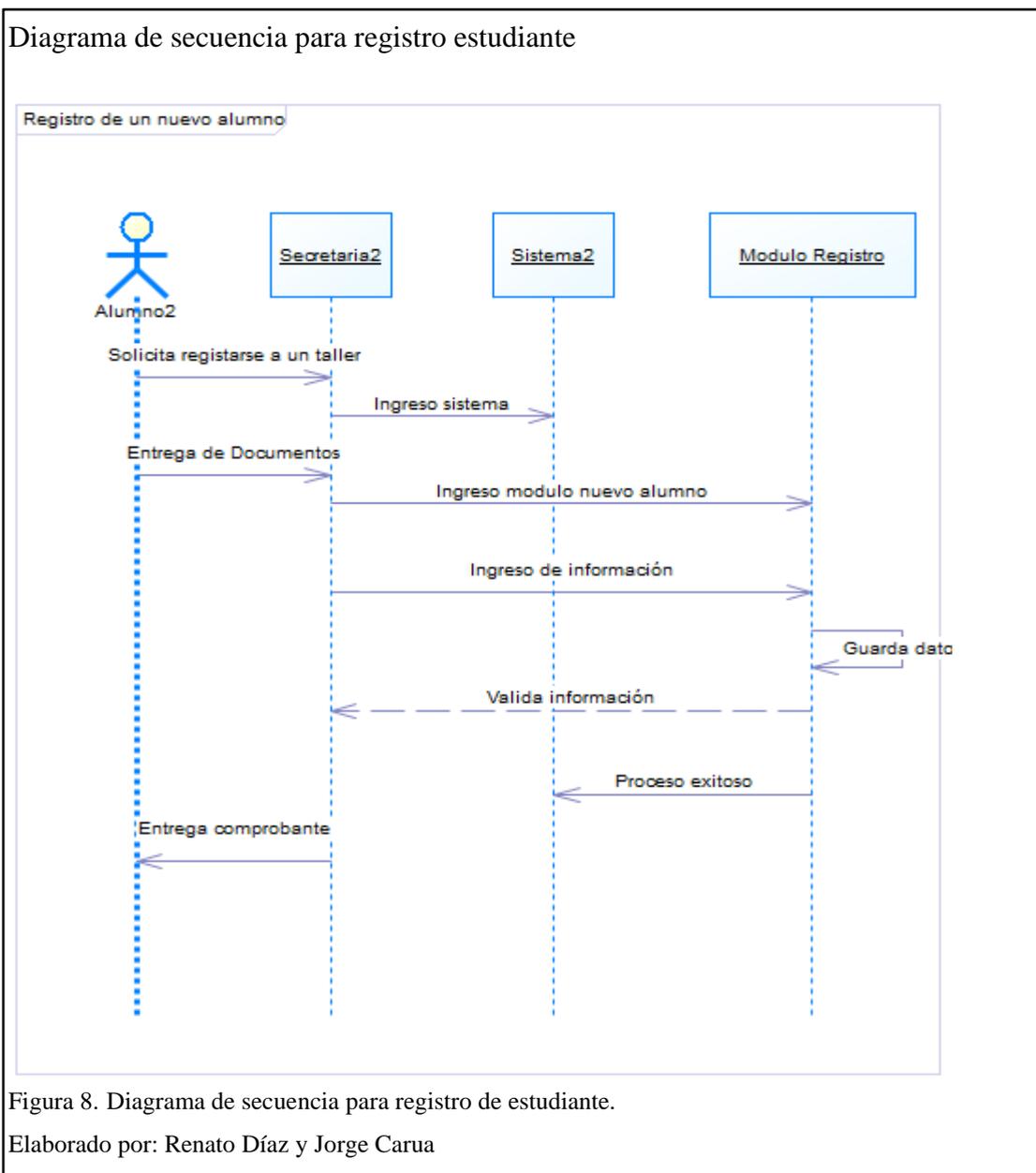


Figura 8. Diagrama de secuencia para registro de estudiante.

Elaborado por: Renato Díaz y Jorge Carua

## Diagrama de secuencia matriculación estudiante

Este diagrama muestra el proceso de matriculación de un estudiante a un taller previa validación en el sistema de la disponibilidad.

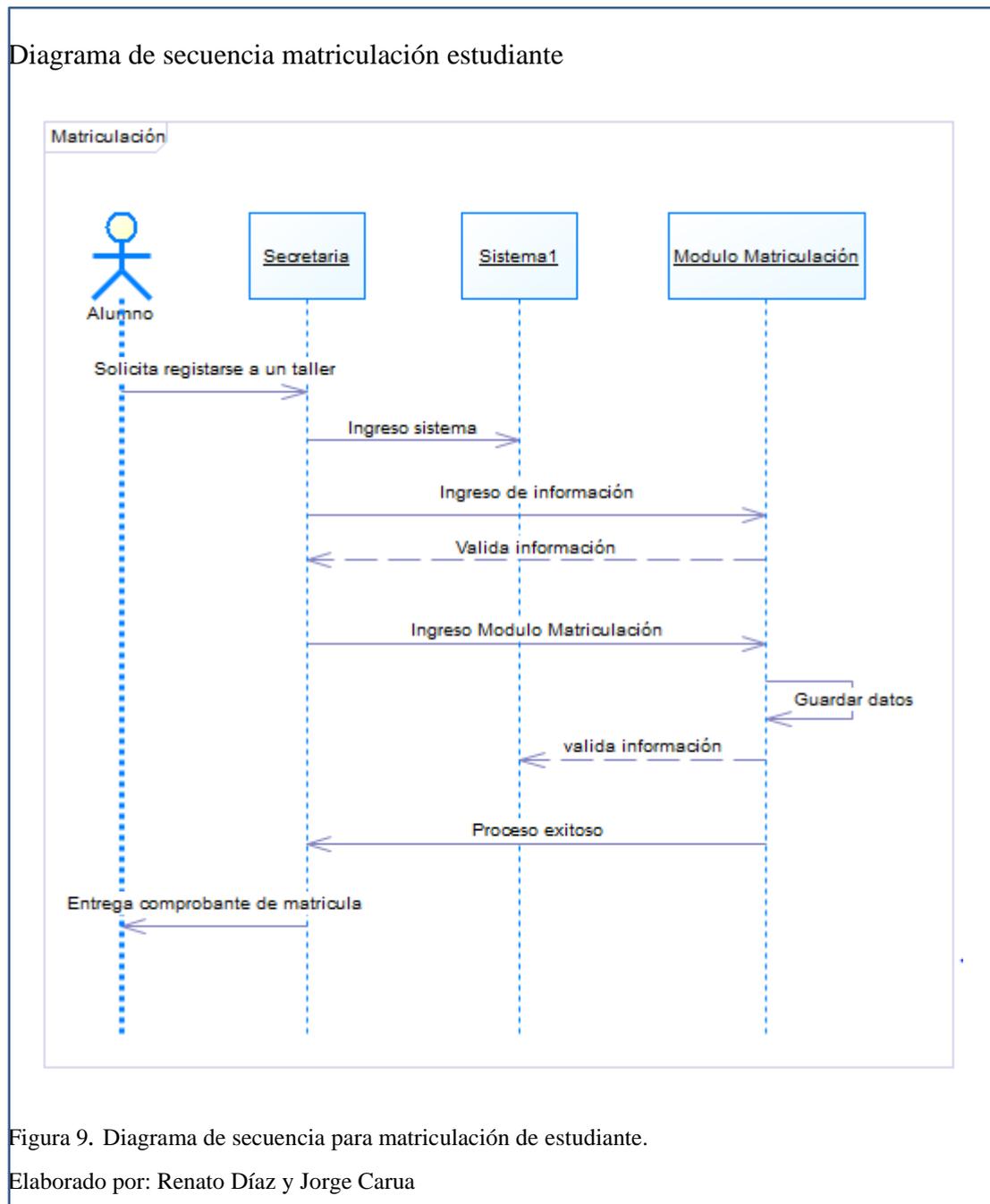


Figura 9. Diagrama de secuencia para matriculación de estudiante.

Elaborado por: Renato Díaz y Jorge Carua

## Diagrama de secuencia asignación horario

Este diagrama muestra el proceso de asignación de horario que realiza el sistema previa validación de la disponibilidad de los datos en el sistema.

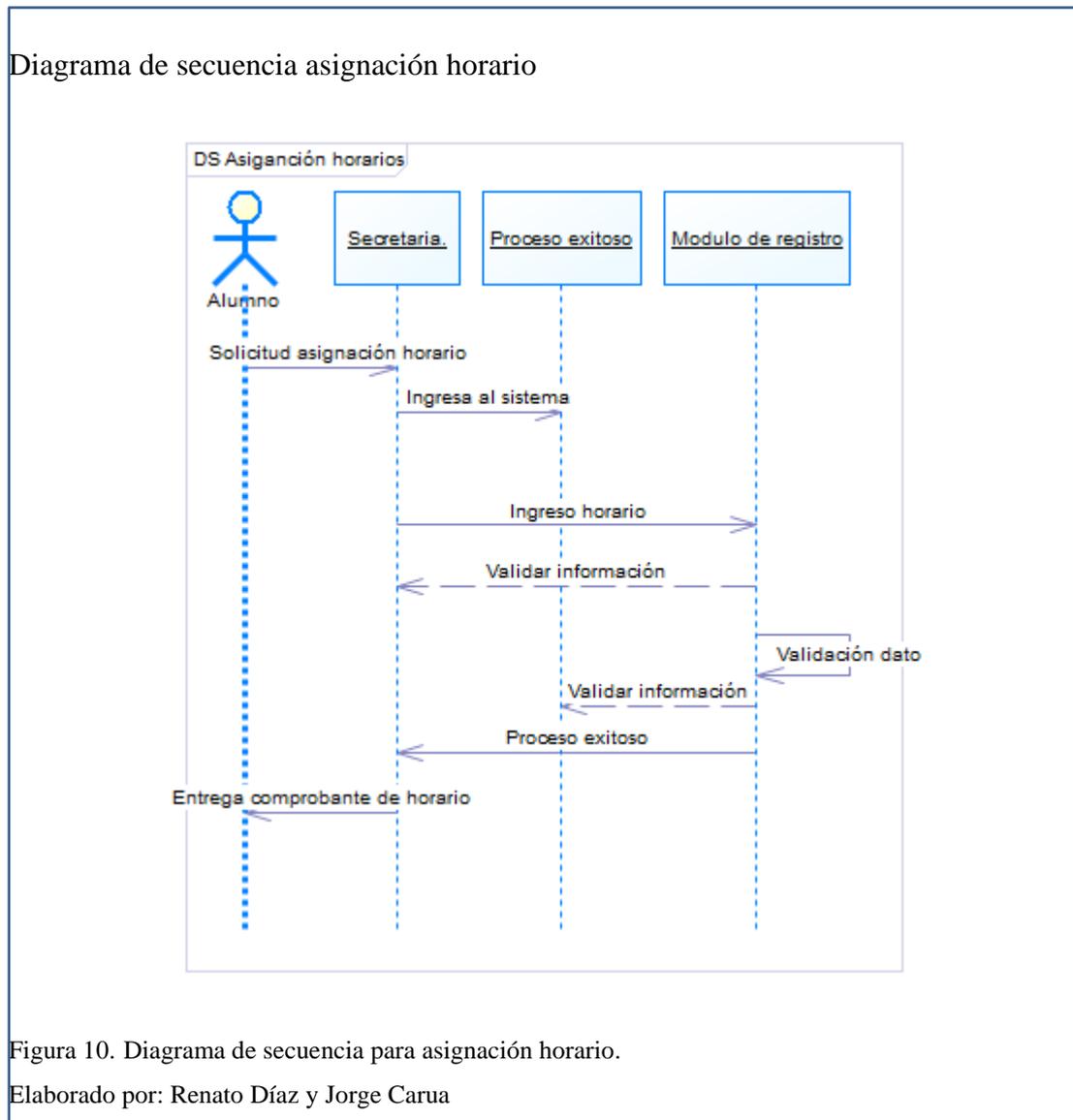


Figura 10. Diagrama de secuencia para asignación horario.

Elaborado por: Renato Díaz y Jorge Carua

## Diagrama de secuencia asignación capacitador

Este diagrama muestra el proceso de asignación de capacitador que realiza el software previa validación de la disponibilidad de los datos en la base de datos.

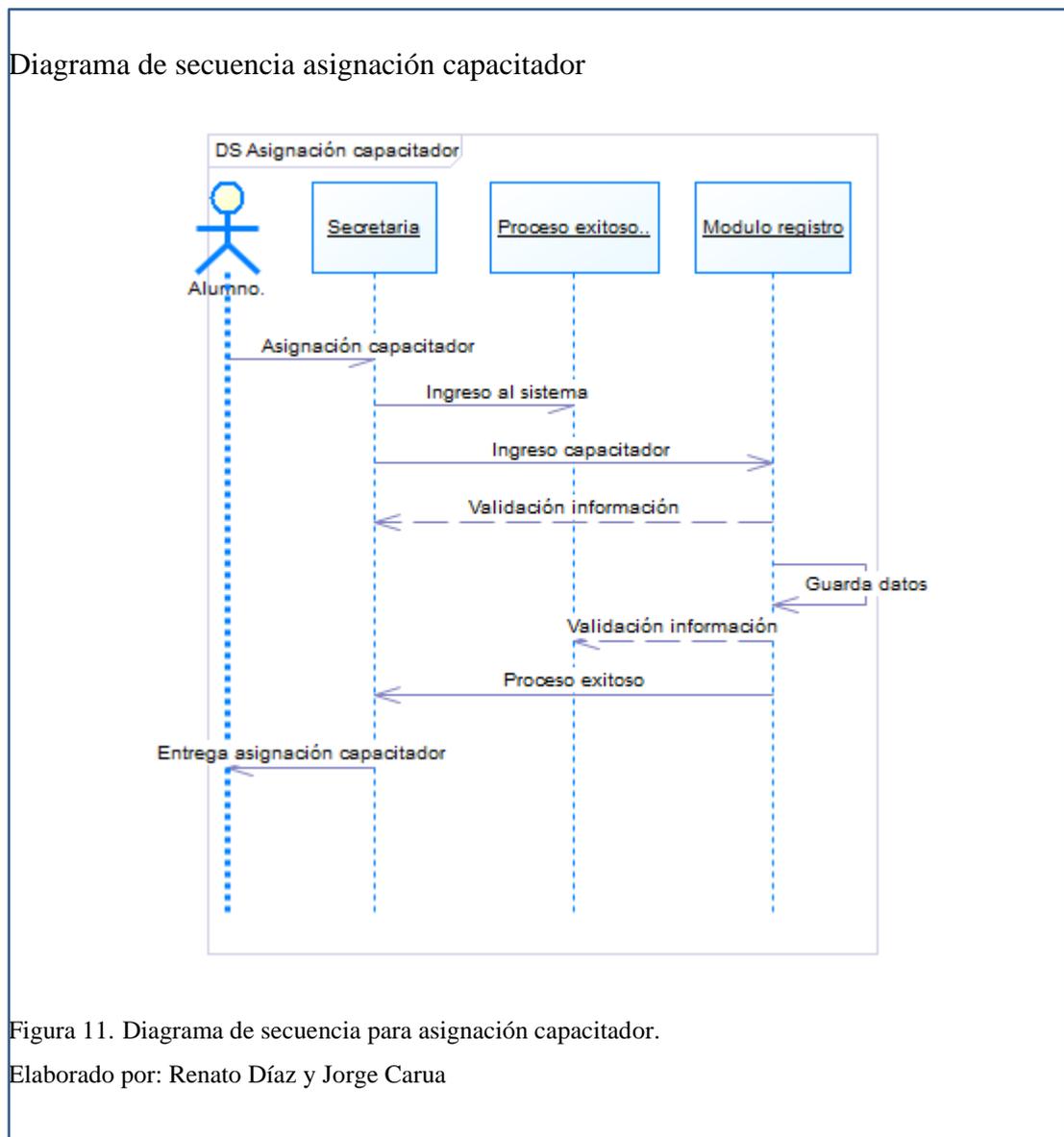


Figura 11. Diagrama de secuencia para asignación capacitador.

Elaborado por: Renato Díaz y Jorge Carua

### 3.3.6 Diagrama de estado

#### Diagrama de estado Registro de estudiante

Este diagrama muestra la secuencia de los estados del proceso de registro estudianto desde su inicio y fin del proceso al ingresar el registro en la base de datos.

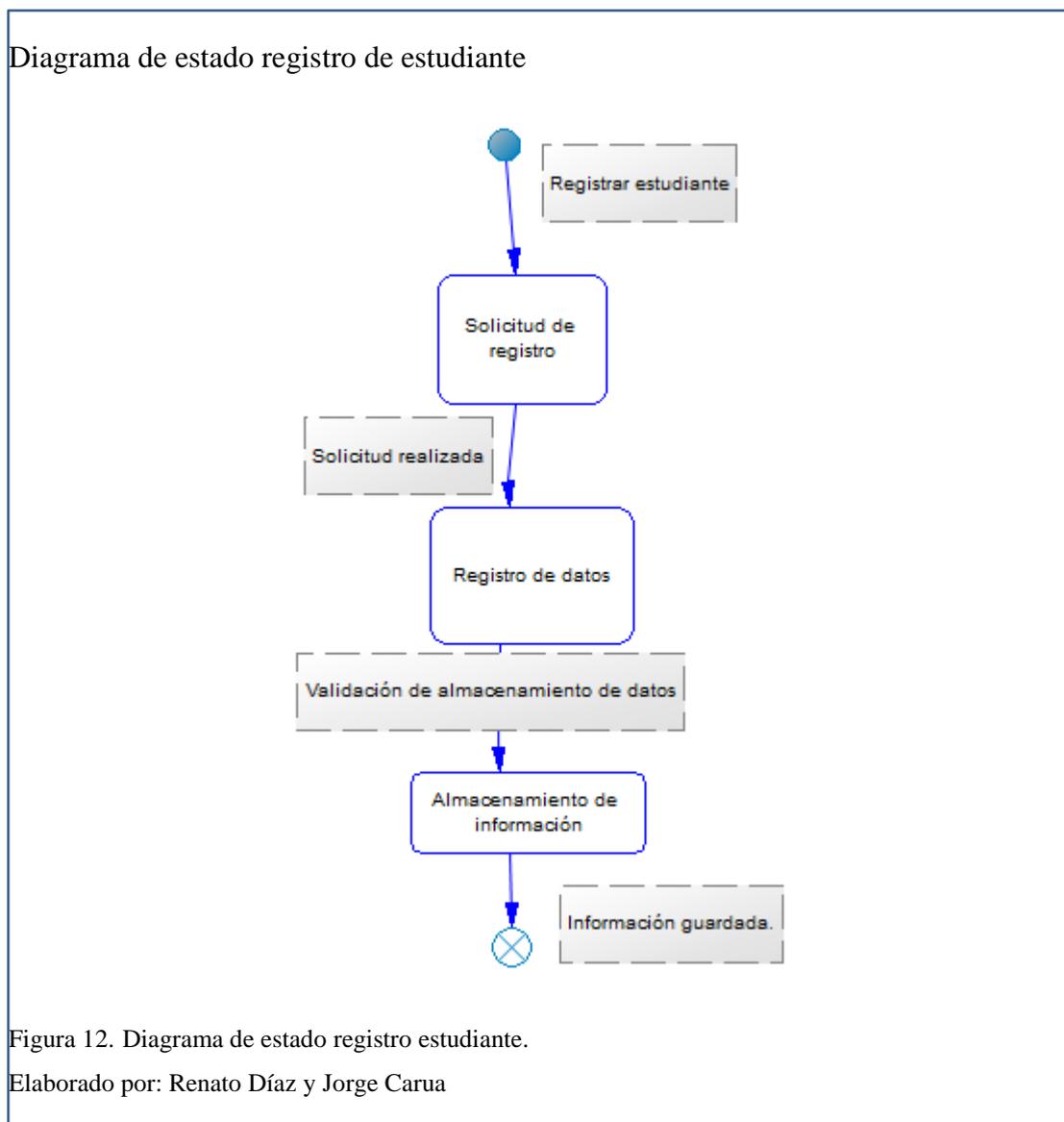


Figura 12. Diagrama de estado registro estudianto.

Elaborado por: Renato Díaz y Jorge Carua

### Diagrama de estado Matriculación de estudiante a taller

Este diagrama muestra la secuencia de los estados del proceso de asignación de taller desde su inicio y fin del proceso al ingresar el registro en la base de datos.

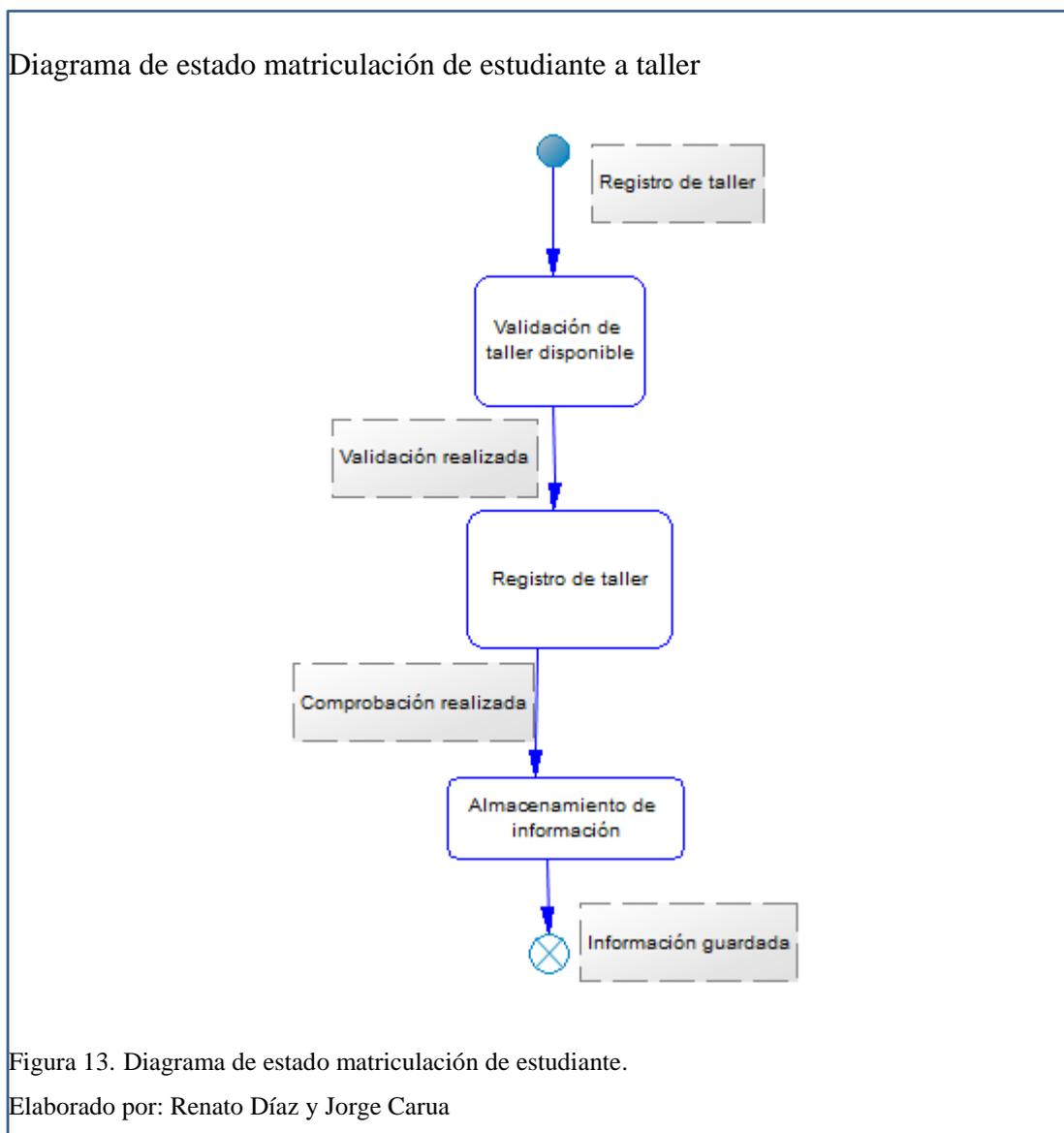


Figura 13. Diagrama de estado matriculación de estudiante.

Elaborado por: Renato Díaz y Jorge Carua

## Diagrama de estado Asignación de horario

Este diagrama muestra la secuencia de los estados del proceso de asignación de horario desde su inicio y fin del proceso al ingresar el registro en la base de datos.

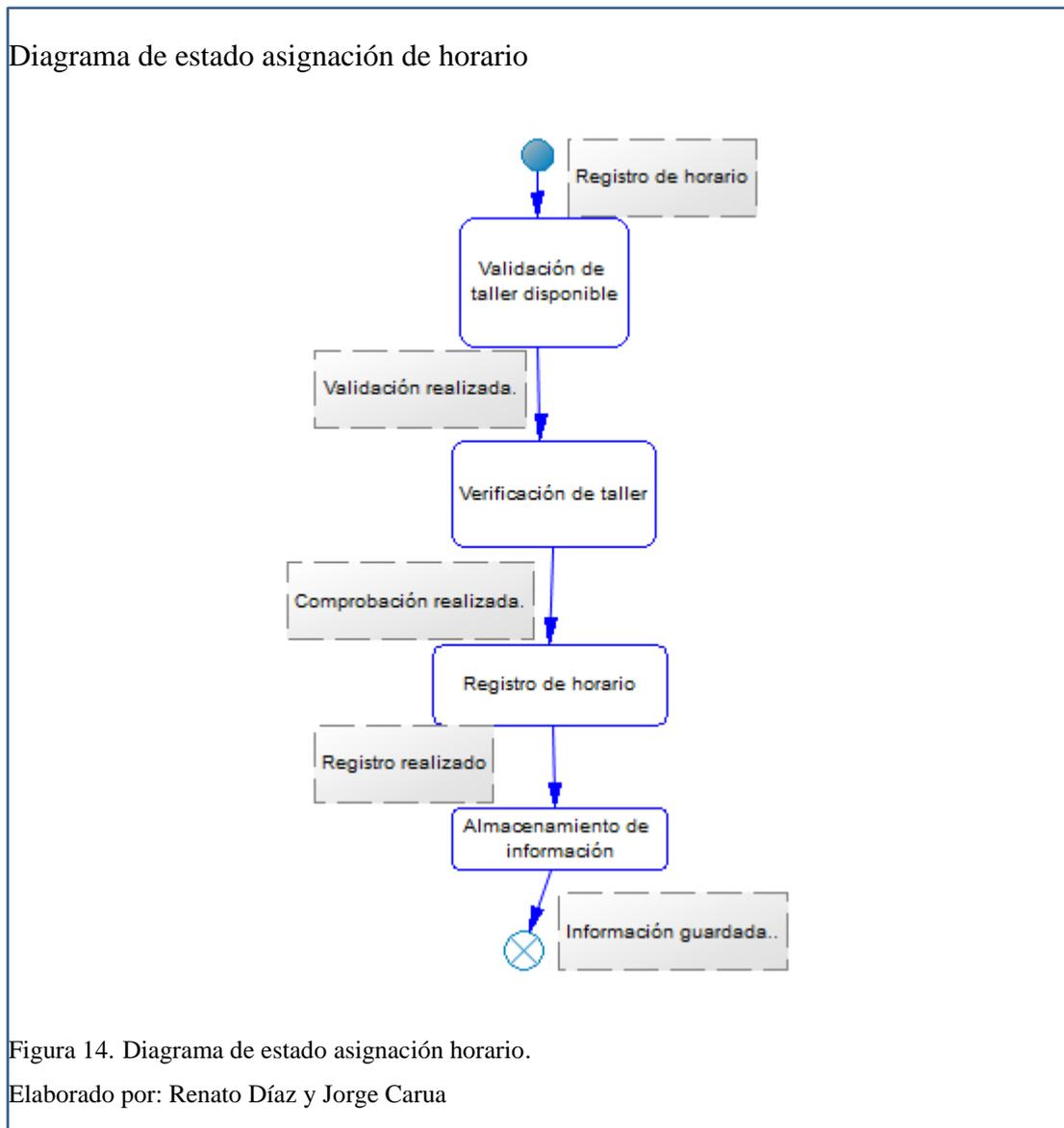


Figura 14. Diagrama de estado asignación horario.

Elaborado por: Renato Díaz y Jorge Carua

## Diagrama de estado Asignación capacitador

Este diagrama muestra la secuencia de los estados del proceso de asignación de capacitador desde su inicio y fin del proceso al ingresar el registro en la base de datos.

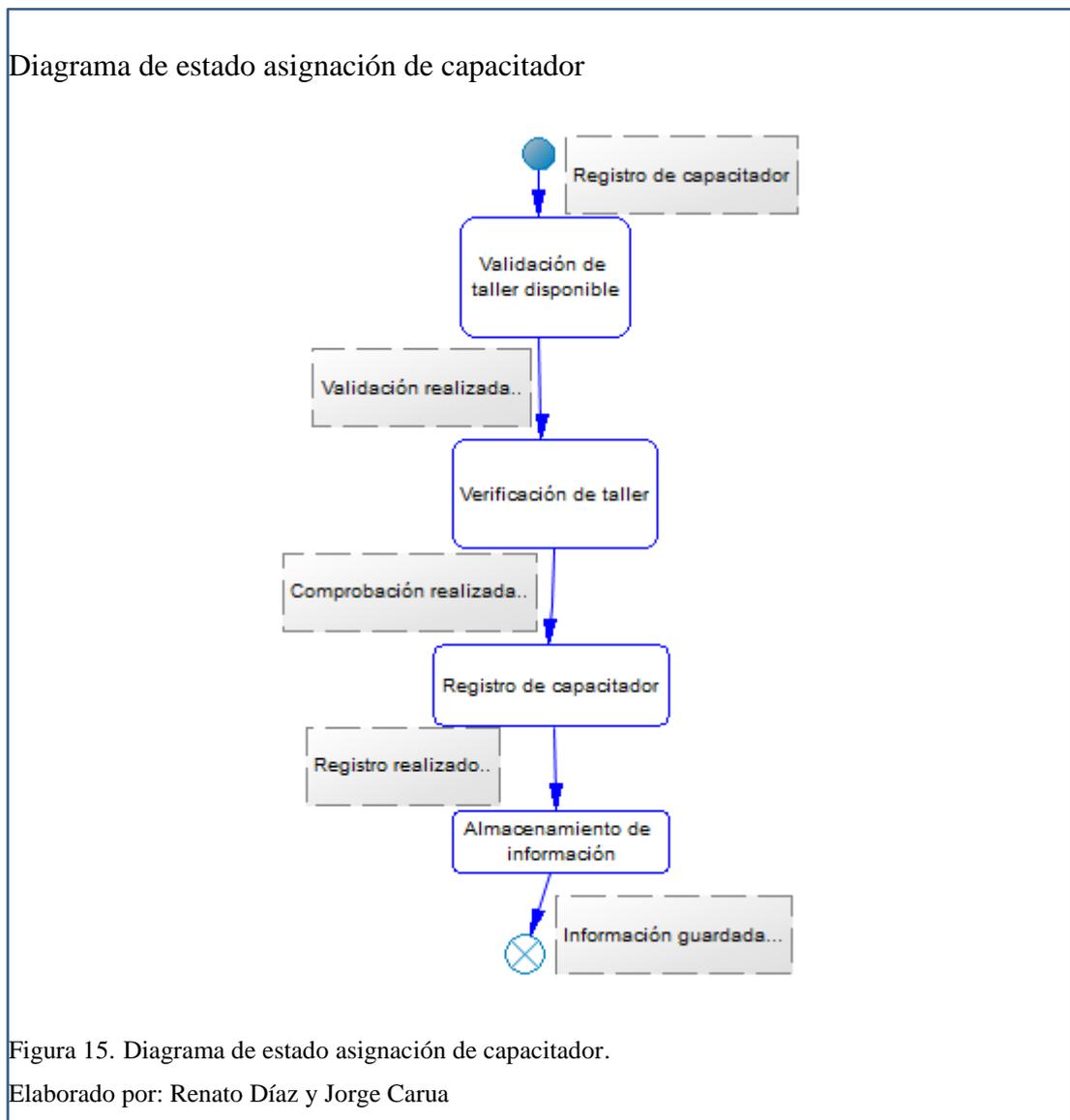


Figura 15. Diagrama de estado asignación de capacitador.

Elaborado por: Renato Díaz y Jorge Carua

### 3.3.7 Diagrama navegacional

#### Diagrama navegacional registro de estudiante

Este diagrama muestra la navegación a las páginas que realiza el proceso de registro de estudiante.

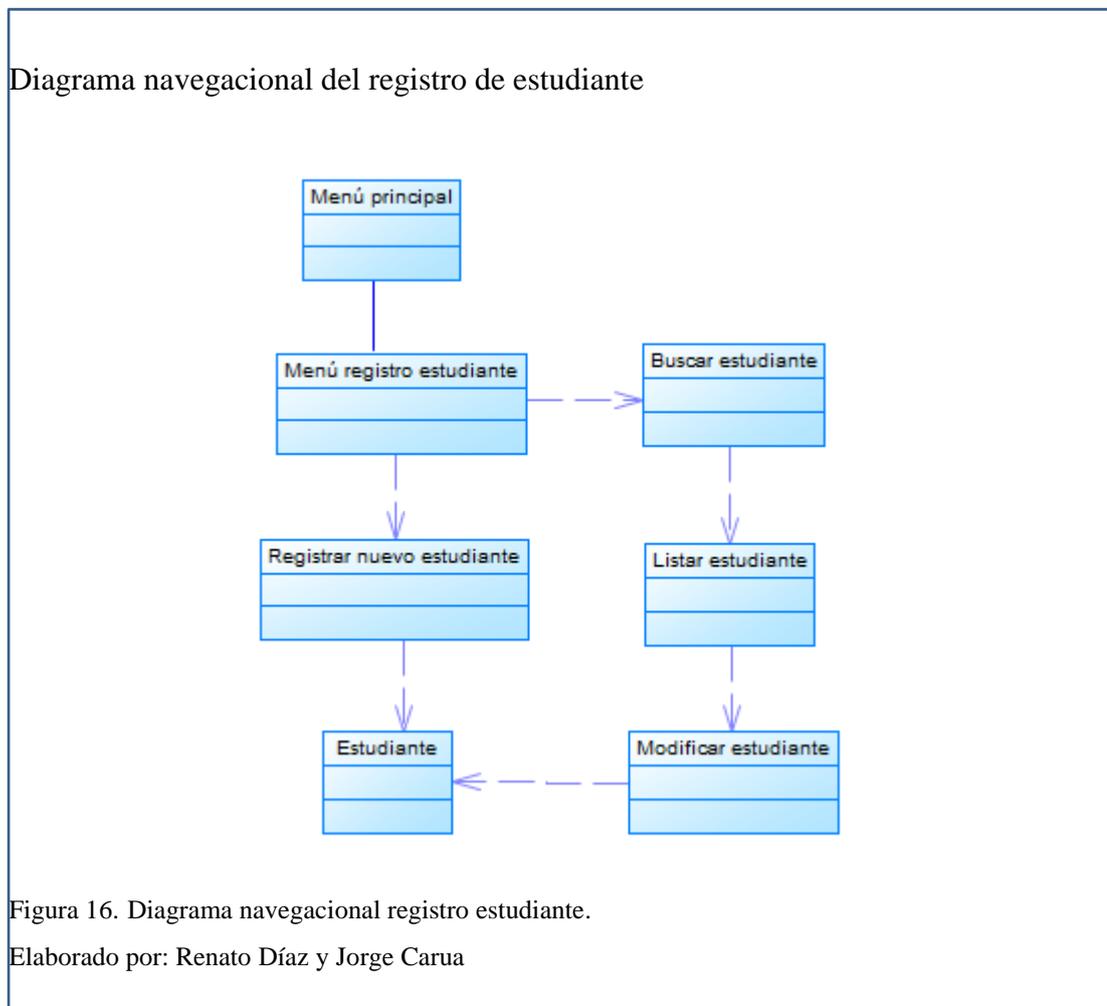


Figura 16. Diagrama navegacional registro estudiante.

Elaborado por: Renato Díaz y Jorge Carua

## Diagrama navegacional matriculación de estudiante

Este diagrama muestra la navegación a las páginas que realiza el proceso de matriculación de estudiante a un taller disponible.

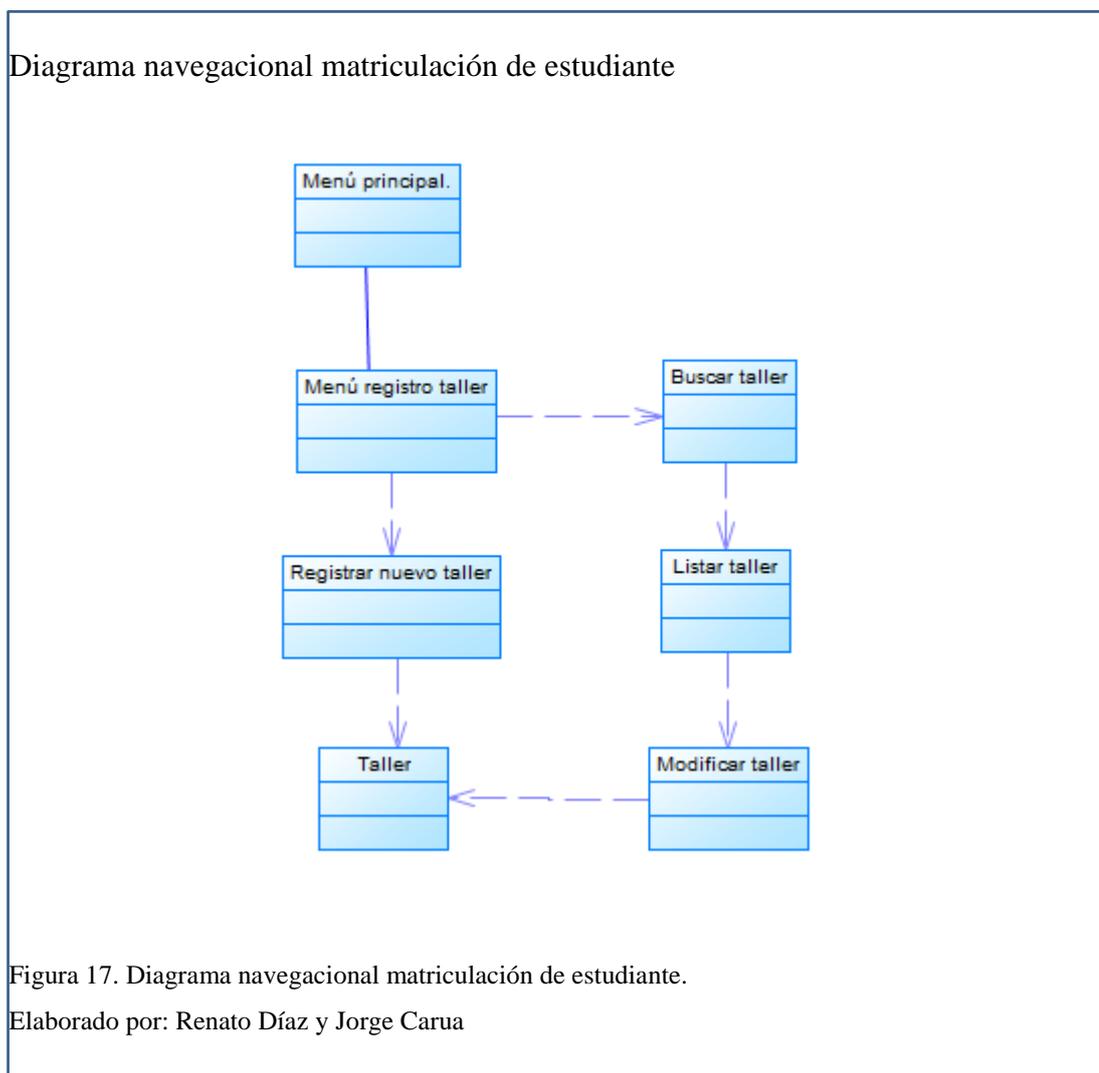


Figura 17. Diagrama navegacional matriculación de estudiante.

Elaborado por: Renato Díaz y Jorge Carua

## Diagrama navegacional asignación de horario

Este diagrama muestra la navegación a las páginas que realiza el proceso de asignación de horario a un taller disponible.

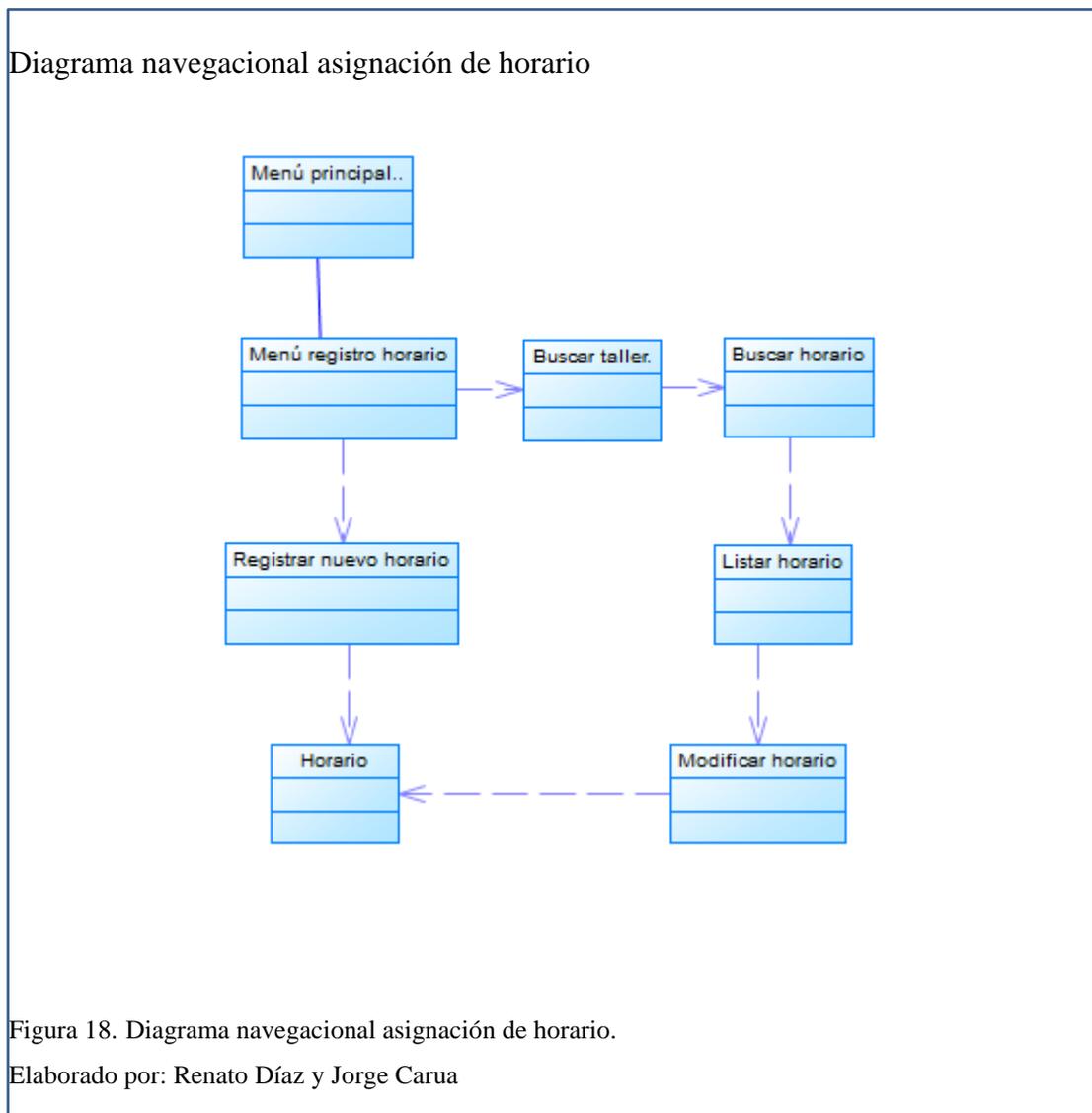
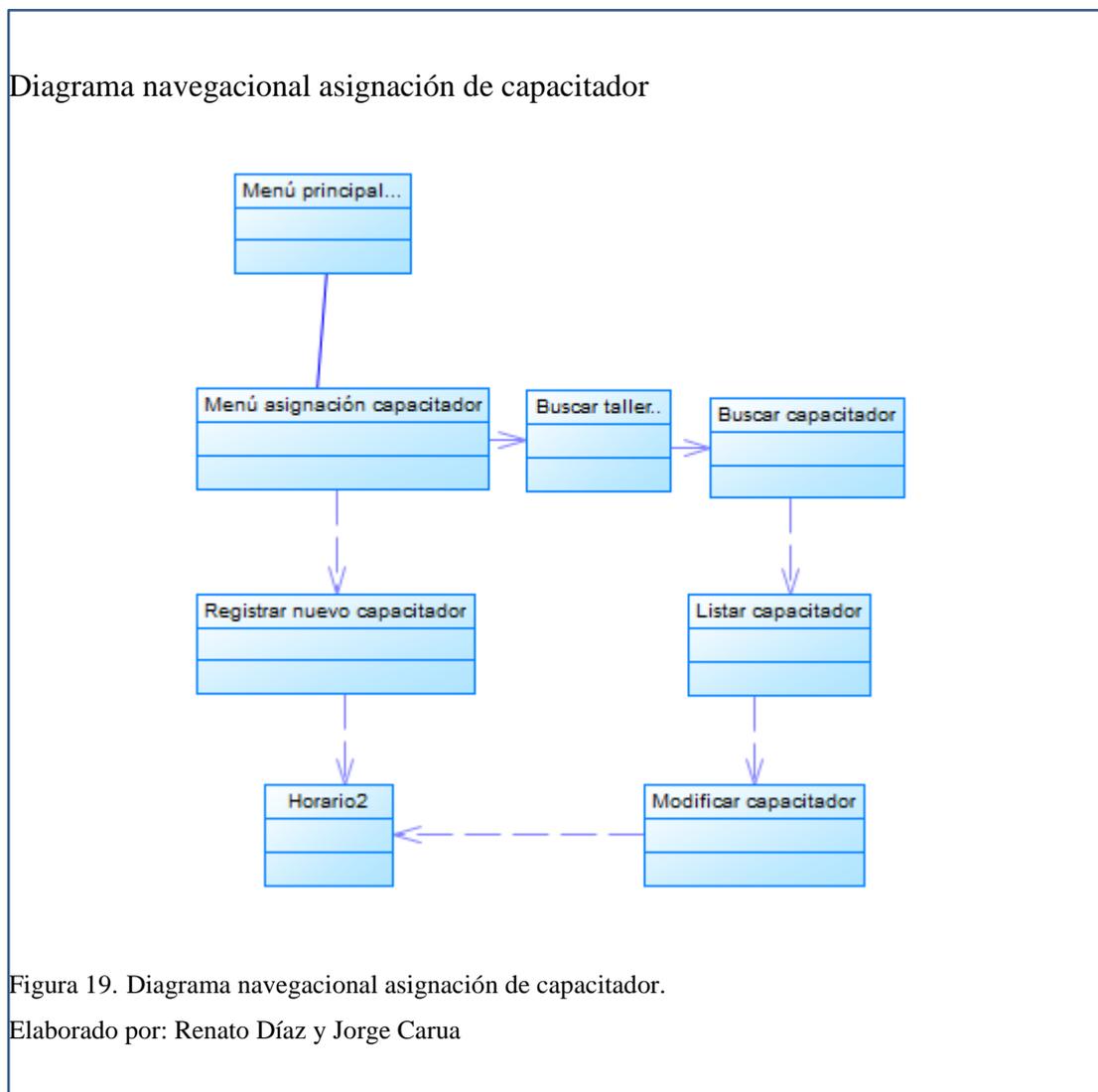


Figura 18. Diagrama navegacional asignación de horario.

Elaborado por: Renato Díaz y Jorge Carua

## Diagrama navegacional asignación de capacitador

Este diagrama muestra la navegación a las páginas que realiza el proceso de asignación de capacitador a un taller disponible.



### 3.3.8 Glosario

A continuación pondremos a consideración de los usuarios los términos que manejamos en el desarrollo del proyecto.

Tabla 4.  
Glosario.

<b>Detalle</b>	<b>Descripción</b>
<b>Administrador</b>	Persona encargada de gestionar el buen manejo y rendimiento del sistema con las revisiones continuas y gestión de los mantenimientos
<b>Secretaria</b>	Persona encargada de administrar la información y gestionar los diversos servicios otorgados por la Federación
<b>Perfil</b>	Característica especial de un usuario que le permite acceder y tomar acción dentro del sistema
<b>Módulo</b>	Conjunto de procesos que se realizan en el sistema
<b>Parámetros</b>	Datos relacionados con los usuarios, talleres, horarios, etc que el sistema maneja para la gestión
<b>Niveles</b>	Se refiere a los talleres y están divididos en 3 niveles: Básico, Intermedio y Avanzado
<b>Taller</b>	Se refiere a la descripción de los cursos que se van a dictar por Capacitador
<b>Periodos</b>	Se refiere a la división por año lectivo
<b>Inscripción</b>	Se refiere al registro de alumnos

<b>Matriculación</b>	Se refiere al registro de alumnos que estarán dentro de los usuarios de la Federación.
----------------------	--

Nota: Glosario del sistema.

Elaborado por: Renato Díaz y Jorge Carua

### **3.4 Aplicación de la metodología Rup en el desarrollo del software**

#### **3.4.1 Fase de inicio**

El propósito es definir y acordar el alcance del software con los usuarios que interactúan con los procesos, identificar los riesgos asociados al software, proponer una visión general de la arquitectura de software y producir el plan de las fases.

#### ***Gestión proyecto***

Al realizar la gestión del proyecto tomaremos en cuenta las necesidades por las cuales se desarrolla el proyecto para la federación basado en la metodología RUP que nos servirá para comprender mejor el análisis como el diseño del sistema.

Durante el proceso de desarrollo llamado Documento Visión se definirá las características del producto a desarrollar, lo cual constituye la base para la planificación. Para el desarrollo del proyecto nos hemos basado en obtener requisitos importantes para la Federación, lo cual nos dará una idea para comenzar el proyecto y durante la Fase Inicio se generará el Documento Visión.

## *Documento Visión*

Es la vista general del proyecto para el desarrollo y diseño de la Fase Inicio, tomando información relevante de la Federación de barrios del Noroccidente de Quito.

La Federación es una institución sin fines de lucro que realiza una serie de gestiones manualmente lo cual implica un rendimiento bajo en sus procesos, por tal motivo, se hace necesario la utilización de sistemas de información.

Se realiza un análisis de las suposiciones y restricciones con respecto del sistema para tener un enfoque de lo que se va a modelar.

- El sistema maneja la gestión de talleres de capacitación pero no realiza funciones de un sistema de facturación o inventarios.
- El sistema maneja información fiable y siempre disponible pero eso no quiere decir que el sistema está protegido del mal uso por parte de los usuarios.
- El sistema está desarrollado para ambiente web con el fin de que en un futuro poder aumentar los servicios.
- El sistema contará con un manejo de perfiles de usuarios (administrador, cliente), donde se especifican las funciones que desempeñarán las distintas personas con la seguridad requerida.

Se describirán los artefactos que ya fueron descritos en la fase de diseño que serán desarrollados y utilizados por el proyecto que constituyen en los entregables, así denominados en el RUP y propuestos para el desarrollo del proyecto.

### ***Modelo Casos de uso***

Este modelo se lo realizará en función a la visión de los actores principales permitiendo determinar al sistema en un contexto organizacional. Este modelo se representará con un Diagrama de casos de uso.

### ***Glosario***

Este documento servirá de guía para los usuarios donde se definirá los términos usados en el proyecto y determinará una terminología consensuada.

### ***Diagrama de secuencia***

Estos diagramas servirán para determinar los objetos como líneas verticales dándonos como referencia el diseño del proyecto y sucesos se determinan de forma horizontal que inician su ciclo desde el emisor al receptor.

#### ***3.4.1.1 Especificación de requerimientos***

Los requisitos del sistema son especificaciones del Documento Visión y serán tomados en cuenta para continuar con el desarrollo y el correcto proceso de la gestión de configuración.

Gestión de configuración será documentada para llevar un registro del status del proyecto, además de la Solicitud de cambio y de las modificaciones que se presenten

para estar siempre disponible a los participantes del proyecto. Para realizar los diagramas de caso de uso del negocio tomamos los actores principales.

- Secretaria: Persona encargada de matricular a los estudiantes.
- Facilitador: Persona encargada de dictar los talleres.
- Alumno: Persona que toma los talleres.

#### Modelo de caso de uso de negocio Secretaria



Figura 20. Secretaria modelo de caso de uso.

Elaborado por: Renato Díaz y Jorge Carua

#### Modelo de caso de uso de negocio Facilitador



Figura 21. Facilitador modelo de caso de uso.

Elaborado por: Renato Díaz y Jorge Carua

## Modelo de caso de uso de negocio Alumno



Figura 22. Alumno modelo de caso de uso.

Elaborado por: Renato Díaz y Jorge Carua

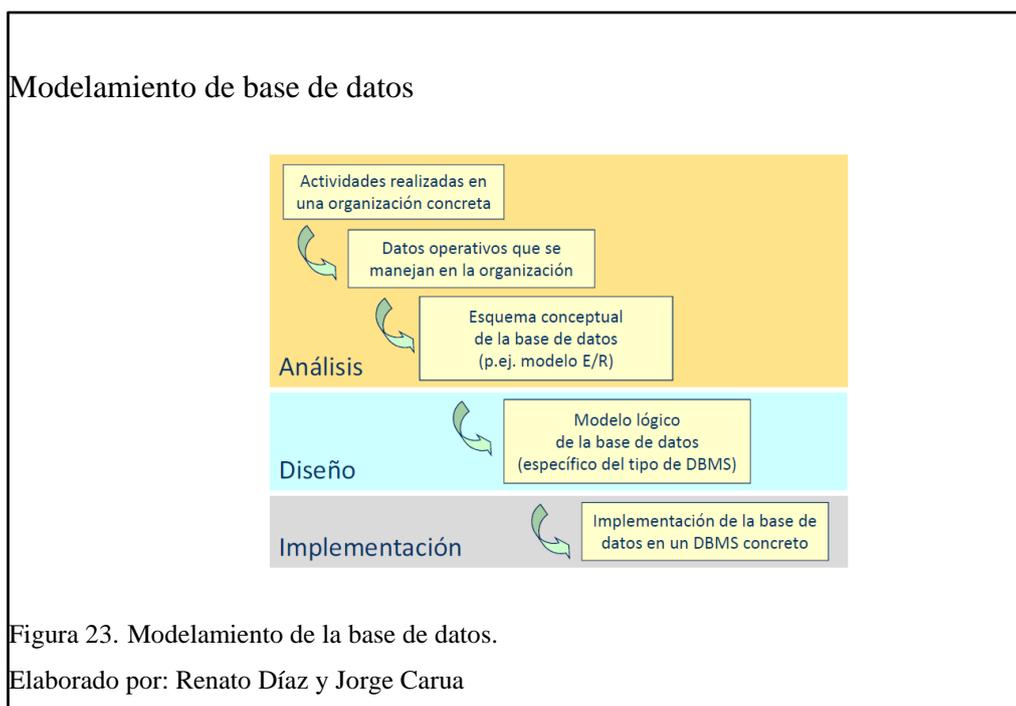
### 3.4.2 Fase de elaboración

Se realiza la especificación de los casos de uso seleccionados, el primer análisis y se diseña la solución preliminar.

#### 3.4.2.1 Modelo de datos del sistema

#### Modelamiento de la base de datos

Una vez establecida las necesidades de la Federación y los usuarios que manejarán el sistema, se inicia con el modelamiento verificando que la base de datos sea eficiente y que cumpla con los requisitos durante el desarrollo. Además debemos tener en cuenta que no sea necesario modificar su diseño. Para lo cual se sigue los siguientes pasos:



## Diagrama conceptual

Para el modelamiento conceptual utilizamos el modelo de entidad/relacional, esta técnica de análisis se basa en la identificación de las entidades y de las relaciones que se dan entre ellas, además permite representar de forma genérica los datos que se pretenden almacenar en una base de datos.

Diagrama Conceptual (entidad/relación)

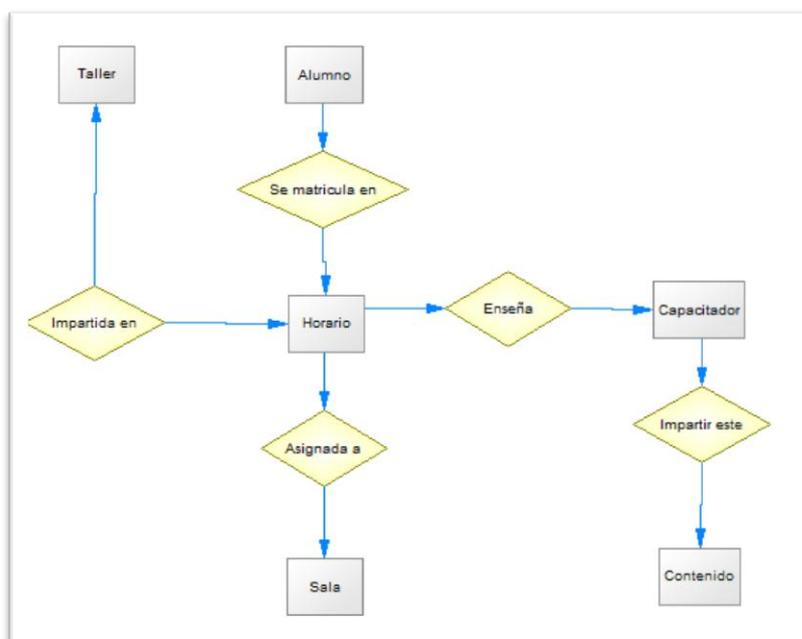


Figura 24. Diagrama Conceptual Entidad / Relación.

Elaborado por: Renato Díaz y Jorge Carua

## **CAPÍTULO 4**

### **CONSTRUCCIÓN Y PRUEBAS**

#### **4.1 Fase Desarrollo**

##### **4.1.1 Interfaz de usuario**

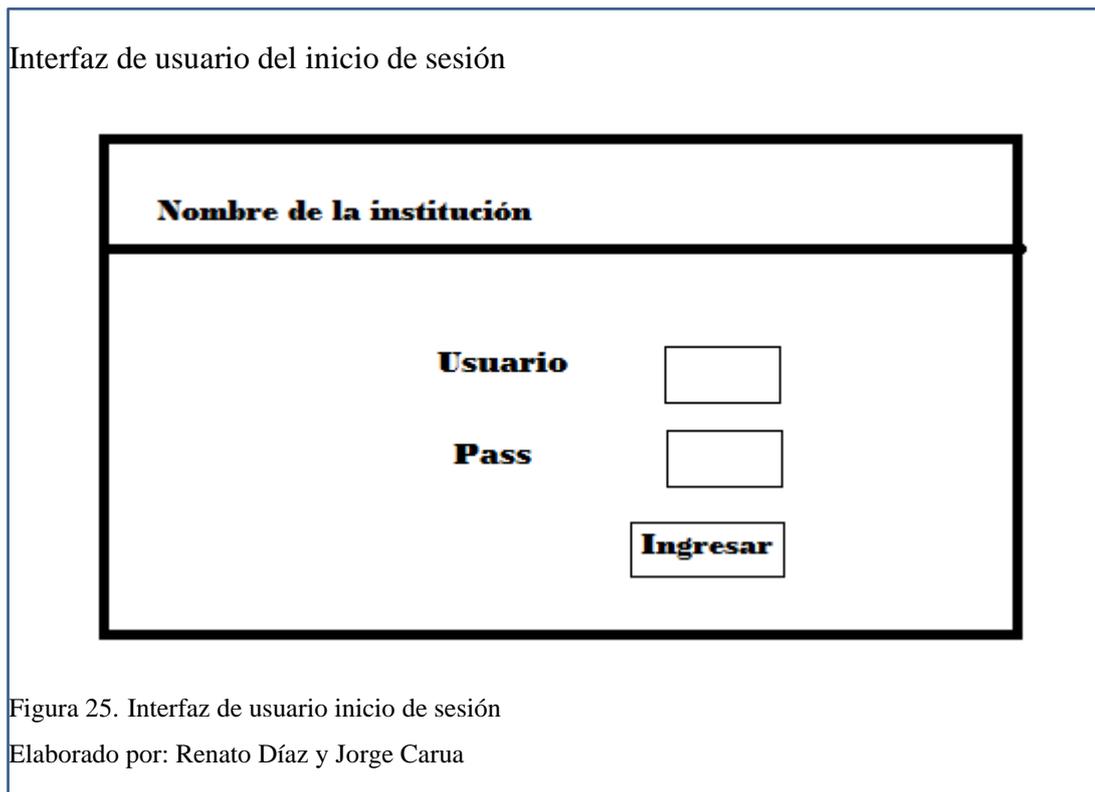
Partiendo de la importancia que tiene la interacción del usuario con el sistema, hemos realizado un diseño que cumpla con las expectativas de la Federación y sus usuarios que manipularán los procesos. Para esto tomamos en cuenta ciertos factores que son:

- Conocimiento del usuario en temas tecnológicos
- Necesidades del usuario en el manejo del software
- El diseño esperado por la Federación
- La información que mostrará el software

Para todo esto proponemos diseñar un ambiente agradable para el usuario que contenga una interfaz estándar, control y libertad de acción de los proceso que realiza el usuario tratando de mantener una estructura similar para los distintos módulos que ofrece el software, intentando no confundir al usuario al realizar las tareas diarias. Para el diseño de las interfaces de usuario estructuramos las pantallas de la siguiente forma: interfaz de inicio de sesión, interfaz estándar e interfaz registros de datos.

##### **Interfaz inicio de sesión**

Interfaz de inicio de sesión tiene la forma rectangular donde la pantalla de dividió en 2 partes para colocar el nombre de la institución en la parte superior y en la parte inferior tenemos 2 textos para ingresar usuario y clave, además un botón de ingreso, ver fig, 25.



### Interfaz estándar

Interfaz estándar tiene la forma rectangular donde la pantalla de dividió en 4 partes para colocar el nombre de la institución en la parte superior, en la parte del medio tenemos el acceso a información, en la parte inferior se dividió en 2 partes de forma vertical, en la parte izquierda tenemos el menú y en la parte derecha tenemos las opciones que se despliegan al tomar un proceso, ver fig, 26.

Interfaz de usuario estándar

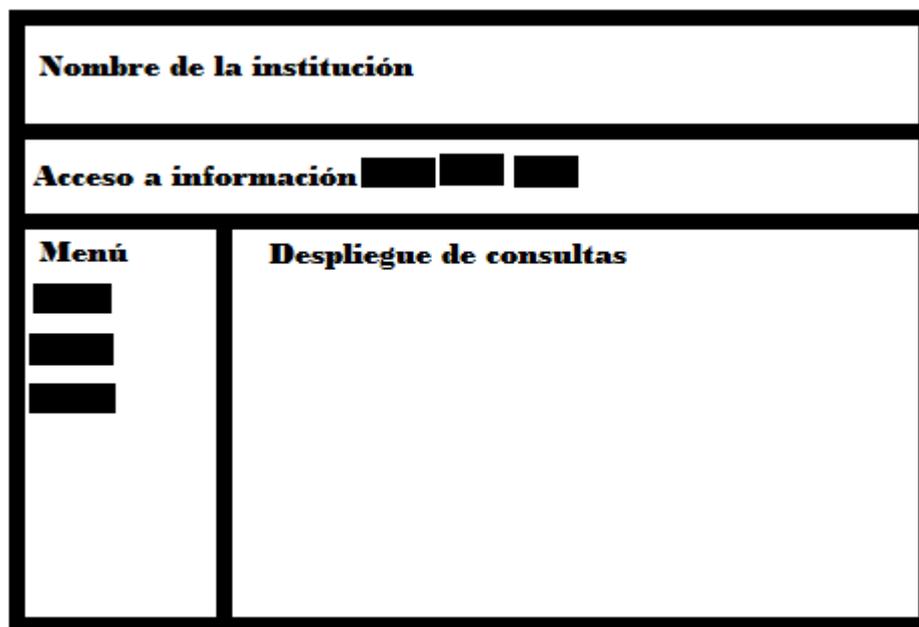


Figura 26. Interfaz de usuario estándar.

Elaborado por: Renato Díaz y Jorge Carua

### **Interfaz registro de datos**

Interfaz estándar tiene la forma rectangular donde la pantalla se dividió en 3 partes para colocar el nombre del módulo al cual se ingresó en la parte superior, en la parte inferior se dividió en 2 partes de forma vertical, en la parte izquierda tenemos las tareas a realizar y en la parte derecha tenemos las plantillas de registro, ver fig, 27.

Interfaz de usuario registro de datos

<b>Módulo</b>	
<b>Tareas</b>  ■ ■	<b>Registro de datos</b>  <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>

Figura 27. Interfaz de usuario registro de datos.

Elaborado por: Renato Díaz y Jorge Carua

## 4.1.2 Codificación

Partiendo del modelo físico se llegó a la construcción de la base de datos, dándonos como resultado el script de la creación del repositorio de los datos del sistema.

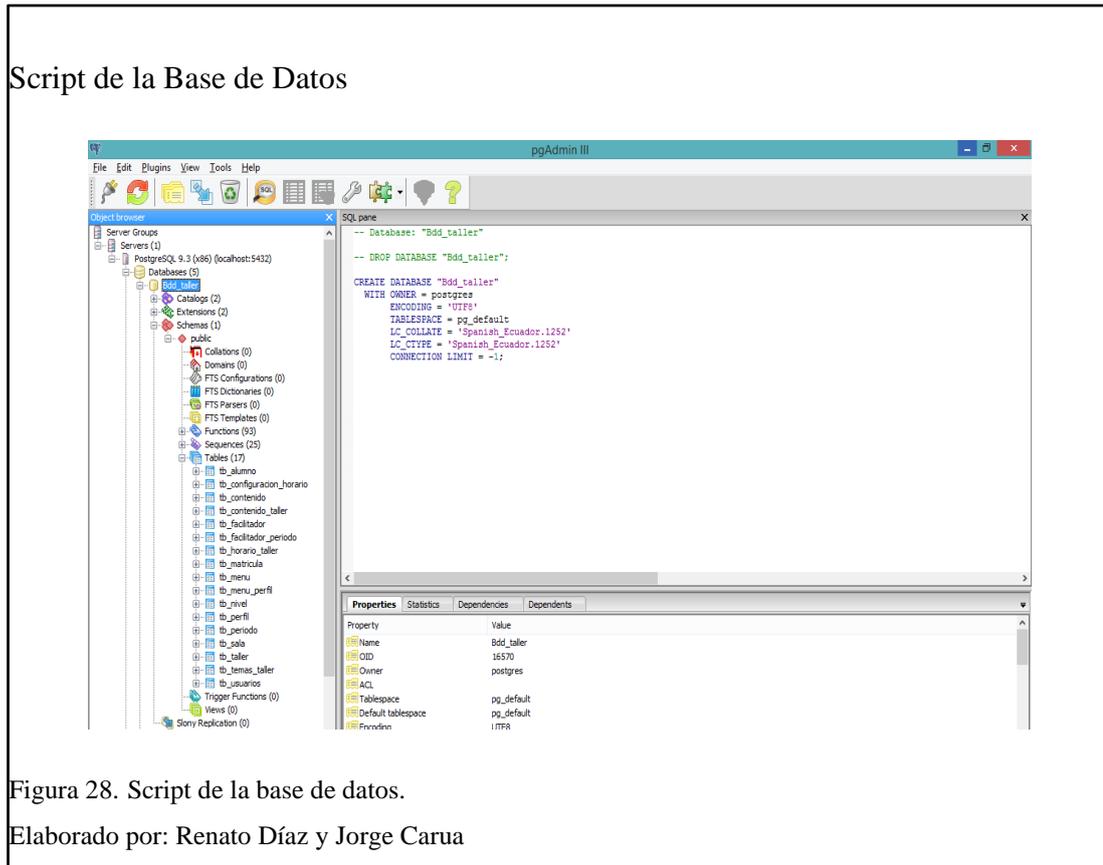


Figura 28. Script de la base de datos.

Elaborado por: Renato Díaz y Jorge Carua

Para la construcción de la base de datos utilizamos los siguientes scripts cumpliendo con el diseño antes ya realizado en el capítulo Fase de diseño.

## Modulo Acceso de usuarios

Este script es utilizado dentro del software para realizar el ingreso de los usuarios dependiendo su perfil, además verifica si existe registro de los datos ingresados en la base de datos del software.

Método:

```
private void ConsultaUsuario(string usur, string pass, string tipo)
{
    NpgsqlDataAdapter adaptadordedatos = new
NpgsqlDataAdapter(NTaller.NegocioTaller.ConsultarUsuario(usur,pass,tipo),
pconexionbdd);
    adaptadordedatos.Fill(tabladedatos);
    if (tabladedatos.Rows.Count > 0)
    {
        foreach (DataRow item in tabladedatos.Rows){
            Session["usuario"] = item["usuario"];
        }
        String url;
        url = NTaller.NegocioTaller.ConsultaUrl(usur);
        Response.Redirect(url);
    }
    else
    {
        lblError.Text = " Por favor revise su Usuario y/o Password";
        Limpiar();
    }
}
```

### **Modulo Matriculación a los diferentes talleres**

Este script es utilizado dentro del software para realizar la matriculación de un usuario con un taller disponible y dentro del año lectivo correspondiente, además verifica si existe registro del taller y la disponibilidad para evitar problemas de cupos no disponibles al ingresar los datos en la base de datos del software.

Método:

```

protected void Button_Gurdar(object sender, EventArgs e)
{
    if (Text_anioest.Text == "" || TextBox_buscarEst.Text == "" || Text_nomestmat.Text
== "" || Text_apellestmat.Text == "" || Text_cedestmat.Text == "" || Text_estmat.Text ==
"" || Text_direcestmat.Text == "" || Text_correstmat.Text == "" || Text_telfestmat.Text ==
"" || Text_celestmat.Text == "" || TextBoxobser.Text == "")
    {
        Lblerror.Text = "Faltan Datos";
    }
    else
    {
        try
        {
            cupos_utilizados = NTaller.NegocioTaller.ConsultaCupos(tallerest, periodoest);
            cupos_asignados = NTaller.NegocioTaller.ConsultaCuposAsignados(tallerest,
periodoest);
            int cu = int.Parse(cupos_utilizados);
            if (cupos_asignados == " ")
            {
                ca = 0;
                ca = int.Parse(cupos_asignados);
            }
            else
            {
                ca = int.Parse(cupos_asignados);
            }
            if (cu >= ca)
            {
                Lblerror.Text = "Este Taller No Cuenta Con Más Cupos Disponibles";
            }
            else
            {
                NTaller.NegocioTaller.IngresandoNuevaMatricula(ci, fecha, periodoest,
tallerest, obser);
                Lblerror.Text = "Estudiante Registrado Correctamente";
            }
        }
        catch { }
    }
}

```

```

        Limpiar();
    }
}
catch (Exception ex)
{
    Lblerror.Text = "Revice información Ingresada";
}
}
}

```

## Modulo Asignación de horarios

Este script es utilizado dentro del software para realizar la asignación de horarios para un taller disponible, dentro de los rangos establecidos, además verifica si está disponible el horario de registro encogido para evitar problemas de cruces al ingresar los datos en la base de datos del software.

Método:

```

protected void ImageButton1_Click(object sender, System.Web.UI.ImageClickEventArgs e){
    Try
    {
        diaregistardo = NTaller.NegocioTaller.ConsultaDiaDisponible(tallerta, dia); //valida si
el dia este disponible para un taller determinado
        horainifin = NTaller.NegocioTaller.ConsultaHoralnicial(dia, hora_inicio, salata);
//valida si esta disponible una hora determinada para el registro de un taller
        sala = NTaller.NegocioTaller.ConsultaSalaDisponible(dia, hora_inicio, salata);//
valida si esta disponible la sala
        totalhoras = NTaller.NegocioTaller.ConsultaHorasTotal(tallerta); //valida si esta
completo las horas para un taller
        horasxtaller = NTaller.NegocioTaller.ConsultaHorasXTaller(tallerta);// extare el
numero de horas a la semana para un taller
        facilitador = NTaller.NegocioTaller.ConsultaFacilitador(facilitadorta); // valida q no
el facilitador no este registrado en un taller
        //valida si esta completo las horas para un taller
        if (totalhoras == "")
    {

```

```

        thoratall = 0;          }
Else      {
    string thor = totalhoras;
    string subthor = thor.Substring(0, 2);
    thoratall = int.Parse(subthor);      }
int horasxtall = int.Parse(horasxtaller);
diaregistardo = diaregistardo.TrimEnd(' ');
dia = dia.TrimEnd(' ');
sala = sala.TrimEnd(' ');
salata = salata.TrimEnd(' ');
facilitador = facilitador.TrimEnd(' ');
facilitadorta = facilitadorta.TrimEnd(' ');

iValidia = string.Compare(diaregistardo, dia, true); //valida si ya esta registrado este dia
iValid = string.Compare(sala.Trim(), salata.Trim(), true); //valida si ya esta asignada la sala
iValidaFacil = string.Compare(facilitador.Trim(), tallerta.Trim(), true); //valida si ya esta
asignado un facilitador a un taller

// valida si ya esta completo las horas
if (horasxtall == thoratall)      {
    Lblerror.Text = "Ya se Encuentra Asignado el Total de Horas";      }
Else      {
    if (iValid == 0)      {
        //valida si ya esta asignada la sala
        Lblerror.Text = "Ya se Encuentra Asignado la Sala a esa Hora";      }
    Else      {
        if (iValidia == 0)      {
            //valida si ya esta registrado este dia
            Lblerror.Text = "Ya se Encuentra Asignado a este Dia";      }
        Else      {
            contenidota = contenidota.TrimEnd(' ');
            if (iValidaFacil == 1)      {
                //valida si ya esta asignado un facilitador a un taller
                Lblerror.Text = "Ya se Encuentra Asignado este Facilitador";      }
            Else      {
                if (horainifin.ToString() != "")      {
                    string h_f = horainifin.Rows[0][1].ToString();

```

```

string h_i = horainifin.Rows[0][0].ToString();
string ho_i = h_i;
string sub_hoi = ho_i.Substring(0, 2);
char[] My_Char = { ':' };
string New_hoi = sub_hoi.TrimEnd(My_Char);
hor_i = int.Parse(New_hoi);
if (hi > hor_i && hor_i < hf)
    {
        Lblerror.Text = "No esta Disponible este Horario";
    }
Else
    {
        if (contenidota == "Practico")
            {
                horatotal = (hf - hi);
                if (horatotal == 2)
                    {
NTaller.NegocioTaller.IngresandoNuevoFacilitadorPeriodo(facilitadorta, tallerta, periodo,
nivelta, salata);
                        Thread.Sleep(1000);
NTaller.NegocioTaller.IngresandoNuevoTemaTaller(contenidota);
                                Thread.Sleep(1000);
                                BloquearDrowlist();
NTaller.NegocioTaller.IngresandoNuevoHorarioTaller(dia, hora_inicio, hora_final);
NTaller.NegocioTaller.IngresandoGriewView(salata, tallerta, nivelta, facilitadorta, dia,
hora_inicio, hora_final, contenidota);
                                        GridViewHorario.DataBind();
                                        Lblerror.Text = "Información Ingresada";
                                        ImageButtAgregar.Enabled = true;
                                        ImageButtNuevo.Enabled = false;
                                }
                            Else
                                {
                                    Lblerror.Text = "Para Horas Practicas solo son dos Horas al Dia";
                                }
                            }
                if (contenidota == "Teoria")
                    {
                        horatotal = (hf - hi);
                        if (horatotal == 1)
NTaller.NegocioTaller.IngresandoNuevoFacilitadorPeriodo(facilitadorta, tallerta, periodo,
nivelta, salata);
                                Thread.Sleep(1000);
NTaller.NegocioTaller.IngresandoNuevoTemaTaller(contenidota);
                                        Thread.Sleep(1000);
                                        BloquearDrowlist();
NTaller.NegocioTaller.IngresandoNuevoHorarioTaller(dia, hora_inicio, hora_final);
                    }
            }
    }

```

```

NTaller.NegocioTaller.IngresandoGriewView(salata, tallerta, nivelta, facilitadorta, dia,
hora_inicio, hora_final, contenidota);          GridViewHorario.DataBind();
Lblerror.Text = "Información Ingresada";          ImageButtAgregar.Enabled = true;
          ImageButtNuevo.Enabled = false;          }
Else          {
          Lblerror.Text = "Para Horas Teoricas solo es una Horas al Dia";
          }    }    }    }    }    }    }    }
catch (Exception ex)    {
          Lblerror.Text = "Revice información Ingresada";    }    }

```

## Modulo Proceso de reportes

Este script es utilizado dentro del software para realizar la validación de reportes de acuerdo a los parámetros seleccionados.

Método:

```

protected void ImageButton1_Click(object sender, ImageClickEventArgs e)    {
          string taller = DropDownTaller.Selected.Value;
          DataTable dt = new DataTable();
          dt = NTaller.NegocioTaller.ConsultarHorarioTalleres(taller);
          ListViewTaller.Items.Clear();
          ListViewTaller.DataSource = dt;
          ListViewTaller.DataBind();    }

```

## 4.2 Pruebas

Es un requisito importante para el desarrollo del proyecto ya que con este proceso podremos medir el desempeño del proyecto, además identificaremos posibles escenarios que en un futuro nos pueden presentar problemas a nivel de manejo de la

plataforma. Con la información obtenida controlaremos los siguientes objetivos plateados para este módulo.

- Identificar el plan de pruebas del sistema
- Identificar posibles inconsistencias del sistema

El plan de pruebas que se aplicará es el método de caja negra el cual será realizado junto a los usuarios para así determinar la funcionalidad del sistema.

### **Prueba de caja negra**

Estas pruebas nos ayudan a obtener escenarios para los cuales enfocarnos a determinar posibles mejoras funcionales del sistema, estas serán realizadas desde la interfaz del sistema. Puntos a tomar en cuentas en las pruebas de caja negra:

- Perfil
- Proceso
- Condiciones de ingreso
- Valor ingresado valido y no valido

Tabla 5.

Prueba caja negra ingreso usuario

<b>Proceso</b>	<b>Condición de entrada</b>	<b>Valor ingresado valido</b>	<b>Valor ingresado no valido</b>
<b>Ingreso usuario</b>	Usuario	Letras, caracteres especiales y números	Campos vacío
<b>Ingreso Contraseña</b>	Contraseña	Letras, caracteres especiales y números	Campos vacío

Nota: Condición de entrada para caja negra.

Elaborado por: Renato Díaz y Jorge Carua

Tabla 6.

Control de prueba ingreso usuario

<b>Condición de entrada</b>	<b>Valor ingresado valido/no valido</b>	<b>Prueba</b>	<b>Resultado</b>	<b>Conclusiones</b>
<b>Usuario</b>	Valido	-jcarua	Ingreso exitoso	Usuario se encuentra registrado en el sistema
<b>Usuario</b>	No valido	-Plopez  -Campo vacío	Por favor revise su usuario y/o password  Falta datos por ingresar	Usuario no se encuentra registrado en el sistema  No se ingresó datos validos
<b>Contraseña</b>	Valido	-1234	Ingreso exitoso	Contraseña se encuentra registrado en el sistema

<b>Contraseña</b>	No valido	-0000	Por favor revise su usuario y/o password	Contraseña no se encuentra registrado en el sistema
		-Campo vacío	Falta datos por ingresar	No se ingresó datos validos

Nota: Condición de prueba para caja negra.

Elaborado por: Renato Díaz y Jorge Carua

Tabla 7.

Prueba caja negra matricula usuario

<b>Proceso</b>	<b>Condición de entrada</b>	<b>Valor ingresado valido</b>	<b>Valor ingresado no valido</b>
<b>Ingreso cedula</b>	Cedula	Solo números, longitud de caracteres 10 dígitos	Campos vacío, caracteres especiales, letras, longitud de caracteres mayor o menor a 10 dígitos.
<b>Ingreso sexo</b>	Caracteres establecidos para sexo	Masculino, Femenino	Campos vacío
<b>Ingreso fecha nacimiento</b>	Año, mes, día	Números con rango establecido	Caracteres, rango distinto a los establecido

Nota: Condición de entrada para caja negra.

Elaborado por: Renato Díaz y Jorge Carua

Tabla 8.

Control de prueba matricula usuario

<b>Condición de entrada</b>	<b>Valor ingresado</b>	<b>Prueba</b>	<b>Resultado</b>	<b>Conclusiones</b>
<b>Cedula</b>	Valido	-1713199550	Estudiante creado	Usuario a sido registrado en el sistema
<b>Cedula</b>	No valido	-123456789	Validar la cedula	No se ingresó datos validos
		-Campo vacío	Falta datos por ingresar	No se ingresó datos validos
		-@#\$\$%^&*	Validar la cedula	No se ingresó datos validos
<b>Ingreso sexo</b>	Valido	-Masculino	Dato correcto	Ingresó datos validos
		-Femenino	Dato correcto	Ingresó datos validos
<b>Ingreso sexo</b>	No valido	-Campo vacío	Falta datos por ingresar	No se ingresó datos validos
<b>Ingreso fecha nacimiento</b>	Valido	2015-03-12	Dato correcto	Ingresó datos validos
<b>Ingreso</b>	No valido	-Campo vacío	Falta datos	No se ingresó

<b>fecha nacimiento</b>			por ingresar	datos validos
		-2015-230-000	Revise datos de fecha	No se ingresó datos validos

Nota: Condición de prueba para caja negra.

Elaborado por: Renato Díaz y Jorge Carua

### **4.3 Fase de transición**

Asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto. (Wikipedia, 2006)

#### **4.3.1 Manual de instalación**

Este manual se puede revisar en los anexo 1.

#### **4.3.2 Manual de Usuario**

Este manual se puede revisar en los anexo 2.

## CONCLUSIONES

Se realizó el estudio de las metodologías para el desarrollo del sistema informático donde se comparó XP con RUP, llegando a la conclusión que la metodología RUP nos fue de mayor utilidad, ya que tiene un proceso de desarrollo iterativo que permite definir los requerimientos en cada fase, en cambio que XP tiene como sistema provocar un cambio constante en todos los requisitos y procesos que debe cumplir la aplicación para su desarrollo.

Se realiza un modelamiento del proceso de asignación de taller y horarios para llegar a la conclusión que la asignación dependerá de la disponibilidad del capacitador, día, hora y la sala asignada, por lo cual se realiza las validaciones necesarias para que el sistema brinde una fiabilidad en su proceso.

Se diseña el sistema informático para la gestión de la Federación de Barrios del Noroccidente de Quito, donde llegamos a la conclusión que este sistema se lo construye en ambiente web para brindar al usuario mayor comodidad y fiabilidad en sus procesos de inscripción y matriculación a los talleres, ya que es una herramienta amigable para el usuario.

## **RECOMENDACIONES**

La información se encuentra almacenada en la base de datos del sistema por lo cual se recomienda a la Federación de Barrios del Noroccidente de Quito realizar respaldos en futuras actualizaciones.

El sistema está creado de forma flexible para poder actualizar sus procesos y se recomienda a la Federación de Barrios del Noroccidente de Quito controlar la asignación de horarios, ya que esto depende de la disponibilidad de varios factores y documentar toda actualización realizada para tener un historial en caso de necesitarlo.

Se tiene el manual de usuario actualizado por lo cual se recomienda a los usuarios revisar el proceso a gestionar para evitar problemas, falta de información, duplicados, etc.

## REFERENCIAS

Buyto. (2009). *Portales Web*. Obtenido de

<http://www.buyto.es/general-diseno-web/que-es-un-portal-web>

Ecuared. (2004). *Sistema Gestor base de datos*. Obtenido de

[http://www.ecured.cu/Sistema\\_Gestor\\_de\\_Base\\_de\\_Datos](http://www.ecured.cu/Sistema_Gestor_de_Base_de_Datos)

José González. (2016). *Tutorial C#*. Obtenido de

<http://dis.um.es/~bmosos/privado/bibliografia/LibroCsharp.pdf>

Lycos. (1994). *Programación Extrema*. Obtenido de

<http://programacionextrema.tripod.com/fases.htm>

Microsoft. (1991). *Visual Studio*. Obtenido de

<https://msdn.microsoft.com/es-es/library/ms173063.aspx>

Microsoft. (1991). *Visual Studio C#*. Obtenido de

<https://msdn.microsoft.com/es-es/library/vstudio/kx37x362%28v=vs.110%29.aspx>

Pmoinformatica. (2012). *Los 5 valores de programación extrema*. Obtenido de

<http://www.pmoinformatica.com/2012/11/los-5-valores-de-la-programacion.html>

UPV. (1993). *Rational Unified Process (RUP)*. Obtenido de

[http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/Model\\_Negocio.html](http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/Model_Negocio.html)

Wikispaces. (2012). *Metodología Rup*. Obtenido de

<https://procesosdesoftware.wikispaces.com/METODOLOGIA+RUP>

Wikipedia. (2006). *Proceso Unificado de Rational*. Obtenido de

[https://es.wikipedia.org/wiki/Proceso\\_Unificado\\_de\\_Rational](https://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational)

## ANEXOS

### Anexo 1. Manual de usuario

#### Inicio del sistema

La pantalla de inicio de sesión se presenta de la siguiente manera:

#### Inicio del sistema – Pantalla principal

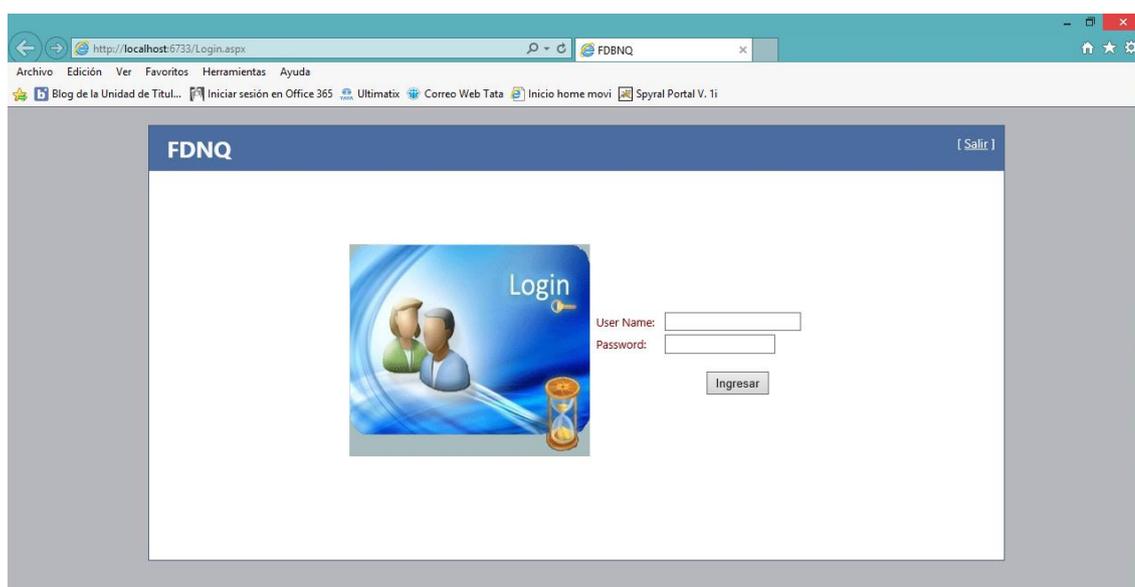


Figura 1. Inicio del sistema.

Elaborado por: Renato Díaz y Jorge Carua

Como se verifica en la Fig 1, debemos ingresar un usuario y password para acceder al sistema, dicho usuario debe estar debidamente justificado por la Federación. Para ingresar damos clic en Ingresar.

## Pantalla de inicio

Una vez que se accede al sistema nos muestra la pantalla de inicio como muestra en la Fig 2, donde podemos realizar la matriculación de los alumnos.

### Pantalla inicio - menú



Figura 2. Menú.

Elaborado por: Renato Díaz y Jorge Carua

Al ingresar a la opción matriculación nos muestra la pantalla de la Fig 3, donde tenemos 2 opciones que son:

- Nuevo estudiante

- Formulario de matricula

## Matriculación estudiantes



Figura 3. Matriculación de estudiantes.

Elaborado por: Renato Díaz y Jorge Carua

## Matriculación de estudiantes

Al ingresar a link [Nuevo Estudiante](#) nos muestra la siguiente pantalla en la Fig 4, donde podremos ingresar un nuevo estudiante con los parámetros solicitados en la plantilla.

## Ingreso estudiante

The screenshot shows a web interface for adding a new student. At the top right is the 'cleo studio' logo. The main heading is 'Ingresar Nuevo Estudiante'. On the left, there is a sidebar titled 'Opciones De Administrador' with two links: 'Regresar' and 'Regresar al Menú'. The main form area is titled 'Nuevo Estudiante:' and contains the following fields:

- Nombres:
- Apellidos:
- # Cedula:
- Sexo:
- Dirección:
- Teléfono:
- Celular:
- Correo:
- Fecha Nacimiento: Año:  Mes:  Día:
- Estado:

At the bottom of the form are two buttons: 'Guardar' (green) and 'Cancelar' (yellow).

Figura 4. Ingreso estudiante.

Elaborado por: Renato Díaz y Jorge Carua

## Asignación de taller

Al ingresar a link Formulario de matrícula nos muestra la siguiente pantalla en la Fig 5, donde podremos ingresar y escoger los datos del taller que vamos asignar al estudiante previa búsqueda en el sistema.

Si el estudiante está registrado en el sistema aparecerán sus datos de forma automática tomando la opción exportar.

## Asignación de taller



### Ficha de Matriculación

**Opciones De Administrador**

[Regresar](#)

[Regresar al Menú](#)

**Nueva Matricula:**

Buscar Estudiante CI:       Buscar:       Exportar: 

Nombres:       Apellidos:

# Cedula:       Sexo:

Dirección:       Correo:

Teléfono:       Celular:

Periodo:       Taller:

Fecha Matriculación: Año:  Mes:  Dia:

Observación:

Figura 5. Asignación taller.

Elaborado por: Renato Díaz y Jorge Carua

## Asignación de horarios

Una vez que el estudiante está registrado procedemos a escoger el periodo lectivo, el taller e ingresamos la fecha de matriculación como muestra en la Fig

Matriculación taller

## Ficha de Matriculación



**Opciones De Administrador**

[Regresar](#)

[Regresar al Menú](#)

**Nueva Matricula:**

Buscar Estudiante CI:  Estudiante Creado Buscar: Exportar:

Nombres:  Apellidos:

# Cedula:  Sexo:

Dirección:  Correo:

Teléfono:  Celular:

Periodo:  Taller:

Fecha Matriculación: Año:  Mes:  Día:

Observación:

Figura 6. Matriculación taller.

Elaborado por: Renato Díaz y Jorge Carua

## **Anexo 2. Manual técnico**

Este manual es desarrollado para la validación del aplicativo Apache Tomcat.

### **Apache Tomcat**

Para tener el servidor disponible es necesario realizar la configuración del Apache Tomcat, brevemente se explica los pasos a seguir.

1. Debemos copiar la aplicación dentro de la carpeta WEBAPP que se encuentra en la unidad C: donde se instaló Apache Tomcat.
2. Verificar la instalación del motor de base de datos de acuerdo a Postgres y su restablecimiento con usuario y clave root.
3. Tenemos que iniciar el servicio Apache Tomcat que se encuentra en panel de control y herramientas administrativas.